目　录

第1章　信号和信号处理

1.1　信号的分类

任何携带信息的物理量都可称为信号。自然界中充满了信号，其中有的是客观存在的，包括自然的和社会的信号；有的则是人类有目的地制造的信号；有的信号是有用的；有的则是不需要的，甚至是有害的，从这些信号中提取愈来愈多的对人类有用的信息，抑制其中有害的信息，是现代信息技术的一个不断追求且永无止境的目标，它推动了信号处理科学在近几十年来的飞速发展。把计算机和数字技术应用到信号处理中，是技术手段上的一个飞跃，它开拓了一个用数字技术来快速、精确地处理信号的广阔天地。因而，数字信号处理就成为人们研究和学习的一个热点。

有各种各样的方法对信号进行分类。

（1）按信号载体的物理特征，可分为：电、磁、光、机械、热、声音等，一些社会科学研究的信号则以统计数据为载体。

（2）按信号中的自变量的数目可分为：

- 一维信号，如语言和音乐，它们是以时间为自变量的。
- 二维信号，如黑白照片或地形图，它们以 x，y 两维坐标为自变量；彩色照片的自变量仍然是二维，但其因变量较多，有红、绿、蓝三色，即三个因变量。
- 三维信号，如黑白电视图像，它的三个自变量是 x，y 坐标和时间 t；彩色电视图像信号则既有三个自变量，又有三个因变量（如红、绿、蓝三色）。
- 四维信号，比如大气中的温度分布，其自变量为经度、纬度、高度和时间。

本书只讨论一维信号的处理问题，它也是研究多维信号处理的基础。

（3）按照信号中自变量和因变量的取值特点可分为：

- 连续时间（Continuous-Time，CT）信号或连续信号，其自变量可连续取值，通常，其因变量也可以在一定动态范围内任意连续取值，即模拟量。所以这样的信号也称为模拟信号，写成 $x_a(t)$。严格地说，自变量的连续性称为连续时间，因变量的连续性，称为模拟信号。物理世界的大多数信号属于这个范围。
- 离散时间（Discrete-Time，DT）信号，虽然时间仍是连续变化的，但变量仅在离散的时刻上定义。大多数离散时间信号是由对连续时间信号采样而得到的，例如，每小时量一次温度，每秒测定十次目标坐标等。但有一些离散时间信号是固有的，比如每天计算一次利息等。因变量 x 的值仍然为可以连续取值的模拟量（在本书中，因主要讨论离散信号，所以对离散信号不加下标，而对连续模拟信号加下标 a）。通常采样周期是一个固定值 T，离散时间是一个顺序增加的整数数组，可写成 nT，其中 n=0,1,…。因此长度为 N 的离散信号 $x(n)$可表示为：

$$x(n) = x_a(t)\big|_{t=nT} \quad n = 0, 1, 2, \cdots, N-1 \tag{1.1.1}$$

- 数字信号（digital signal）：为了在数字系统中对这些因变量数据进行存储和处理，就必须把它们用字长较短的二进制数来表示，这个运算称为量化(quantize)。得到二进制数的最低位所代表的值，就是分辨率，又称为量化步长。如果离散信号中的因变量不是模拟量而是量化了的数字量，或原来的模拟量经过了取样和模拟/数字变换，输出为离散值，则这种自变量和因变量都只取离散值的信号称为数字信号。把数据 x 变成为量化值 x_q 的数学变换关系如下。

$$x_q = \Delta x \cdot \mathrm{round}(x/\Delta x) \tag{1.1.2}$$

假如量化步长为 Δx，则 $x/\Delta x$ 是把数据 x 量化为步长的倍数，一般情况下，它是带小数的。round($x/\Delta x$)是用四舍五入的方法取此倍数为整倍数。再把取整后的倍数乘以 Δx，就复原为量化后的值 x_q。

这个量化算法，不仅用于单个数据，也适用于数组。把数取整的方法还有向上取整[x]、向下取整[x]、向零取整等，相应的 MATLAB 函数是 ceil,floor,fix 等。round 是最常用的。

用 MATLAB 来实现时，由于它只接受英文字符，设 deltax=Δx，量化语句就写成：

```
xq=round(x/deltax)*deltax;
```

例如，若 x=[3.14159,0.38455,54.3753,-0.134273];deltax=2^-4;
则执行量化语句 xq=round(x/deltax)* deltax;的结果为：

```
xq =    3.1250    0.3750   54.3750   -0.1250
```

这是一种按绝对值或定点数进行量化的方法。对所有的数，量化步长都是 2^{-4}=0.0625。所以其量化造成的绝对误差最大为 deltax/2。如果考虑相对误差，即数的误差与其绝对值之比，可表示为：rerror=(xq-x)./abs(x)，得到如下结果：

```
rerror =-0.0053   -0.0248   -0.0000   -0.0691
```

结果是：对大的数，量化造成的相对误差小；而对小的数，量化造成的相对误差大。对于一个信号序列，由于只能用同一个 A/D 转换器进行量化，量化步长必须是相同的，所以只能采取这种方式。人们希望的往往是使相对误差大体相等的量化方法，也称为浮点数的量化方法。这可以用软件实现，用硬件是不容易做到的。第 9.2 节中将介绍用非线性变换的方法来实现对信号的近似同等的相对量化。

下面举个例子说明这三种信号，以加深理解。

例 1.1.1 设连续信号为 $x_a(t)=\sqrt{t}+\cos t$，$0\leqslant t\leqslant t_f=6$。

其中 t_f 为终点时刻。若采样周期为 0.5s，求采样后的离散时间信号；又设 A/D 变换的量化步长为 0.5，求量化后的数字信号，并绘出曲线。对连续时间信号、离散时间信号和数字信号进行比较。

解: 用 MATLAB 语言来描述此问题的模型。首先建立连续时间的自变量向量 t，程序中，自变量 t 是可在 0～t_f 秒中取任何值的连续量，因变量 x_a 也可以取任何值的模拟量，但计算机无法严格地表示连续量和模拟量。就自变量来说，它只能把 t 取得很密。程序中将其表示为 t=0:dt:tf；把时间增量 dt 取得很小，例如取 dt=0.001。够不够密不能只看 dt 绝对值的大小，而要看相邻两个样本值之间变化是不是足够小，真正严格的数学标准只有通过傅里叶分析才能建立。因变量就取 MATLAB 中的双精度格式，其动态范围是 10^{-308} 到 10^{308}，而有效值是十进制数 15～16 位，分辨率是 2.24×10^{-16}，看来很接近于模拟量了。即使如此，严格地从理论上说，程序中的 t 和 x_a 相对于真正的连续量和模拟量还差很远，而且这个距离是计算机永远无法超越的。工程上只能不断逼近理论值，这是学习本课中应具备的工程观点。

将此时间轴按 $T = 0.5$ s 的采样周期进行分割，得到的离散时间的自变量向量 n。通常它从零开始，取顺序增加的整数，直到终点 t_f/T，表示为 n=[0:1:6/T]。将按这个采样周期对此信号 x_a 进行采样，所得的离散时间信号的表达式为

$$x(n) = x_a(t)\big|_{t=nT} = \sqrt{nT} + \cos(nT)\text{，}\quad n=0, 1, \cdots, 12$$

它们在时间轴上只能在间隔为 T=0.5s 的间断点上取值，而作为因变量的 x 在纵坐标方面的取值是连续的，即仍然是模拟信号。它取的是 MATLAB 中的双精度数据格式，因而和 x_a 取值相同，画在图上，它们应该是重合的。

将 x 量化为 x_q 的方法已在上面详细说明过了，此处只要设定 deltax=0.5 即可。

根据以上分析，建议读者用笔算的方法求出 n=0～12 共 13 点处的离散因变量 x 和它们的量化值 x_q，并且可以用草图画出曲线。然后与下面的 MATLAB 程序 hc111 比较，研究用 MATLAB 编程有哪些优点。

```
dt=0.001; tf=6; t=0:dt:tf;                  % 建立连续自变量向量
xa=sqrt(t)+cos(t);                           % 原始的连续时间信号 xa(t)
T=0.5;n=0:tf/T;                              % T 为采样周期，建立离散自变量向量
x=sqrt(n*T)+cos(n*T);                        % 采样周期为 T 的离散时间信号 x(n)
deltax=0.5;                                  % deltax 为 x 的量化步长
xq=round(x/deltax)* deltax;                  % 舍入量化后的数字信号 xq
subplot(1,2,1),
plot(t,xa,':'),hold on,grid on               % 画出连续时间信号曲线
plot(n*T,x,'o')                              % 画出离散时间信号曲线
stem(n*T,xq,'*'), grid on                    % 画出画出数字信号曲线
legend('连续信号 xa','离散时间信号 x','数字信号 xq')  % 画出图例标注
subplot(1,2,2)
stairs(n*T,xq),    grid on                   % 画出数字信号采样保持后恢复的连续信号曲线
legend('将数字信号采样保持','恢复的连续信号曲线')   % 画出图例标注
set(gcf,'color','w')                         % 将本图的背景色置为白色
```

本程序最后加上了完整的绘图语句，其中包括加图例标注的 legend 语句和设置背景色的 set 语句。这是本书的第一个程序，所以详细一些。为节省篇幅，在后面的程序中将会精简多数的绘图和标注语句，读者可从网上（www.broadview.com.cn）提供的程序集中下载完整的程序。

程序运行的结果见图 1.1[①]。图 1.1（a）给出了这三种不同信号的曲线，连续时间模拟信号的曲线用虚线画出，离散时间信号用空心圆点画出，而数字信号用星号画出，图上还画出了纵横坐标量化后的网格。为了看清楚时间离散化和因变量量化对原始波形的影响，现把数字信号在一个采样周期内展宽，画在图 1.1（b）中。可以看到它与（a）图所示的虚线差别很大，这是因为取了特别大的采样周期 T 和特别大的量化步长 Δx 的缘故。如果把这两个量不断减小，则在一个采样周期内展宽后的数字波形就会很接近于原模拟波形。读者可以自行减小程序中的 T 和 deltax，重新运行并进行验证。这里也要提示读者注意学习本程序的编程技巧，应当使一些可调整的变量只在程序前面出现一次，改变它的值时只需改动一个数据，仍能保证程序的完整执行。

① 本书中的插图是由程序运行直接产生的，MATLAB 认为正、斜体是一样的，其图中外文字一律为正体，不遵循“物理量名称的外文代号为斜体”的规则。

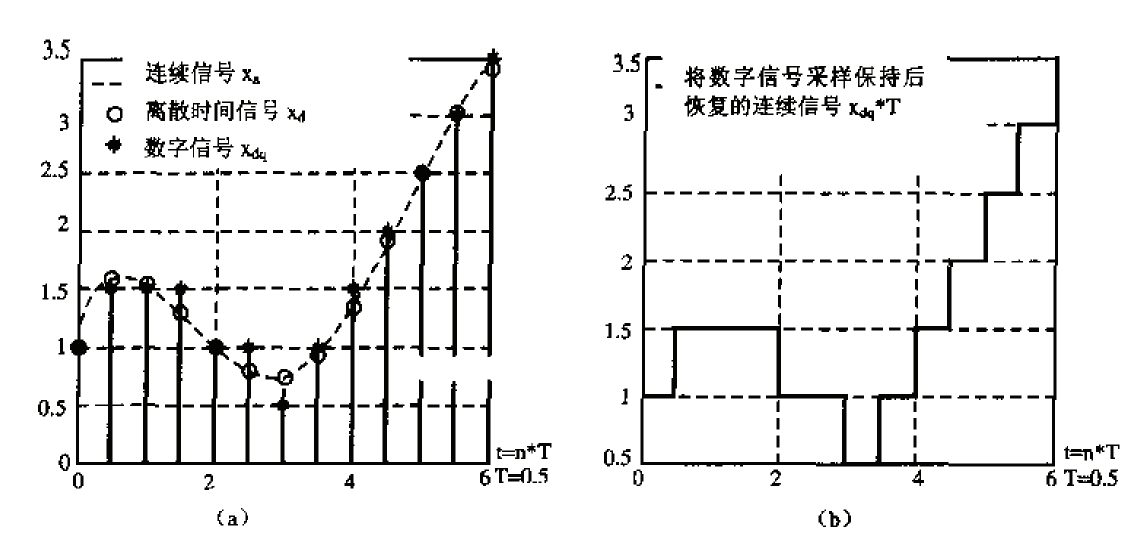


图 1.1.1　连续信号、离散时间信号和数字信号

（4）按照信号的来源来分，那就太多了，这里只能举些例子，如下所示。

- 心电（Electro Cardio Graphy，ECG）信号：其形状见图 1.1.2。波形的特征点由 P，Q，R，S，T 来表征，从它们的间距、幅度和形状来确定心功能的状况，所以它以时域分析处理为主。信号的幅度约 1μV，而脉冲的宽度约为 0.02s，因此要求放大器的带宽约在 100 Hz 以上，才能避免明显的波形失真。因为信号很微弱，心电图仪很容易受 50Hz 的市电影响，要采取多种信号处理措施来消除市电干扰。

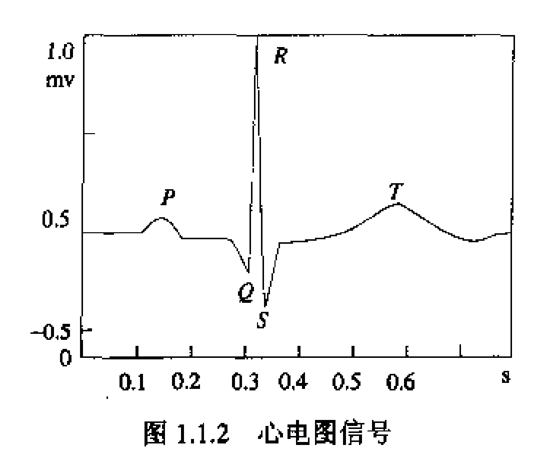


图 1.1.2　心电图信号

- 脑电图（ElectroEncephaloGram，EEG）信号：它的幅度范围为 2mV～100mV，频带从 0.5 Hz 到 100 Hz,又分成 5 个区段：0.5 Hz～4 Hz 为 delta 段，4 Hz～8 Hz 为 theta 段，8 Hz～13 Hz 为 alpha 段，13 Hz～22 Hz 为 beta 段，22 Hz～30 Hz 为 gamma 段。由于每一段的信号分别反映不同的大脑活动，所以要把各频段的信号区分开来，频域分析和滤波是主要的信号处理手段。
- 地震信号：它由岩石的自然运动或人为的地下爆炸等产生。主要有三种弹性波，其中两种在地球球体中传播，而第三种则沿地球表面传播。因为地球中岩石的运动是三维的，因此，地震仪要在三个方向安装传感器，分别测出和记录这些信号。地震信号除了对于研究地球和地震有用之外，很重要的一方面是用来找石油和天然气。因为油和气的力学特性与固体不同，在地震信号中可以得到反映，提取这些信号就可以得到地下存储的石油和天然气的信息。
- 发动机信号：内燃机、汽轮机、航空发动机、水力涡轮机等各种机械，在转动中会产生丰富的振动信号，其中包含了它们的运行信息，例如，转速、驱动进程、动力大小、异物干扰等，从中可以监视它们的运行过程，并在故障出现的早期就提出警告。
- 语言信号：人们研究语言信号，早期是为了语言的传送，比如电话和它的多路传送，后来又为了语言的合成与识别。语音的频带大约为 300 Hz～3000 Hz 就可以听懂了，

但其质量比较低。高质量的语言传送，语言的合成，对语音频带要求就比较高了。特别是语音识别，对语言信号的分析和特征提取，提出了更高的要求。

- 音乐信号：音乐信号的频带通常低至 20 Hz，高至 18 000 Hz，有些特殊的乐器甚至还超过了这个范围，因此在信号处理时，要考虑表现各种乐器特殊音色的要求，还要考虑到音乐厅的回声、混响和立体声等的效果。
- 统计数据：信号可以来自自然现象，也可能来自社会现象。大量的社会经济的数据，其中包含着丰富的表征社会发展和缺陷的信息，也需要人去提取。所以信号处理的许多原理已经在社会科学研究中得到应用。

1.2　信号处理

1.2.1　信号处理的举例

对于物理世界存在的信号，要增强或提取它的某些成分和特征，又要抑制和消除它的另一些成分。为此而采取的措施都可以称为信号处理。在计算机或数字技术出现之前，所采用的都是，也只能是模拟的方法，即物理的方法，举例如下。

- 汽车在崎岖的道路上行驶，车内的人和物将受到地面起伏所造成的冲击和颠簸。人们利用车的轮胎、支架弹簧及液压阻尼器等，可以吸收和过滤其中强的冲击尖峰和高频率的振动。同样的机械减震原理也可用在其他车辆、船舶和飞行器中。
- 由于交流电电流经过全波整流生成的直流电源包含着 100Hz 和更高次的谐波。因此在其后必须加上由电感、电容、电阻组成的模拟滤波器，把直流保留下来，并把交流成分抑制到足够低的水平。
- 在电声系统中，要使扬声器放出频谱均匀的强功率声音，扬声器的结构必须进行非常讲究的设计，还要合理搭配多种喇叭，以保证它对各个频段有均匀的响应。为了在礼堂和音乐厅中有良好优美的音色和混响，在建筑形状、吸音板布局和材吸音料的选择上都要有周密的分析和计算。
- 采用物理原理工作的滤波器，有晶体滤波器和声表面滤波器等。前者利用晶体的电—机械谐振特性产生非常方陡的滤波通带，后者则是利用声表面波的振动传播特性，产生迟延和滤波。
- 光学处理的各种反射镜、透镜、光栅、滤光片等。它们可以在光的运动方向、极化方向和光谱范围等多方面对光信号进行分析、选择、过滤和增强处理。
- 微波中的长线，阻抗匹配器、负载吸收器等，可以对高频信号按频率、极化形式等进行选择，并分别予以处理；天线则可以控制电磁波发射的方向，也可以说是进行高频信号的空间处理，其结果是增强某些方向的电磁场，减弱其他方向的信号。
- 电话的双频拨号：过去的电话拨号是靠脉冲计数来确定 0~9 这 10 个数字。不仅拨号的速度慢，而且没有办法扩展电话的其他服务功能，所以现在都改为双音拨号。它的原理如下：每一个数都由低带和高带两个音频混合而成，低带频率有四个：697 Hz，770 Hz，852 Hz 和 941 Hz；高带频率为 1 209 Hz，1 336 Hz，1 477 Hz 三个（另一个 1 633 Hz 备用）。三个高带和四个低带频率可以组合成 12 个数，如图 1.2.1 所示。中

继站利用信号处理技术，在短促的拨号音中迅速分辨出它的高低两个频率，从而确定它代表的数。不仅速率快，而且在0到9之外还有两个多余的码 # 和 * ，可以用来扩大电话的服务功能。

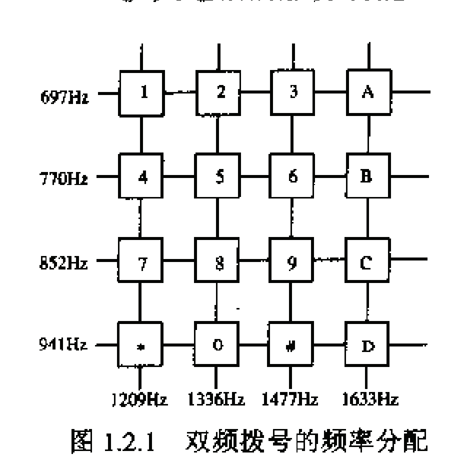


图 1.2.1 双频拨号的频率分配

上面举出的这一些例子，可以说明信号处理所涉及领域之广泛。本书所讨论的信号处理的数学分析方法是通用的，可以适用于上述所有例子。但真正进行处理，特别是进行数字化处理，目前还只限于弱电和低频领域。其中的许多情况必须使用物理的方法。随着技术的迅速发展，这个“低频”的上界不断向上扩张，现在已经由音频发展到视频，而且有些其他物理领域的信号处理问题，也通过传感器变换为电信号来处理，然后，再变换回该物理领域去。不过这种变换有些是难于做到的，所以要看到数字信号处理有局限性的一面。

1.2.2 信号处理的一般任务

信号的时域分析处理——一类是进行信号分析，比如说分析心电路中各波形的时间间隔、峰值的大小和宽度、周期性、形状的变异等。时域的分析也用在对雷达回波的分析，从背景和干扰中提取目标的坐标和速度，等等。另一类是生成和变换各种调制信号，对信号的动态范围进行压缩扩张，用门限方法进行噪声的抑制，等等。前一类是属于时域中信号的线性处理，主要通过信号的加减、时移、倍乘、卷积、求相关函数等来实现。而后一类，则属于非线性的变换和处理。比如调制过程就是非线性的，在本书中只在第9章中涉及一些讨论非线性信号处理的过程，前八章中只进行线性分析处理。

信号的频域分析处理——比如前面谈到的脑电波的分析，要分别找到各频段的脑电波幅度来判断大脑的活动；在电声系统中，大量地应用频域分析的技术、高保真的放大器，就要通过各种滤波器来提高某些频段的放大量，抑制另一些频段的放大量；还要通过延迟和多路的方法，实现立体声和混响效果，音乐的录制则更为专业，它是先用多路的传感器和磁轨把乐队演奏时的各个乐器的声音分别录下来，然后各自做适当的滤波处理后，再以不同比例加以延迟和混合，造成在音乐厅中产生的混响效果，最后把合成信号制成磁带或光盘。

信号的频域和时域分析通常是要结合在一起进行的。例如，要不失真地反映心电图的时域波形，放大器就要有足够宽的频带，同时要能抑制 50Hz 的市电干扰，放大器在市电频率附近要有陷波特性。因此就需要从频域分析着手。

滤波器设计——在对信号进行分析的基础上，需要设计适当的系统，以便能提取出有用的信号，而抑制噪声干扰。按照这样的目的来设计的系统也就是滤波器，滤波器的设计通常是在频率域进行的。模拟滤波器设计技术已经相当成熟，本书要用到它，但将着重讨论数字滤波器的设计问题。

对线性领域来说，信号处理有两大任务：一是信号分析，其重点是频谱分析；二是滤波器设计分为有限长脉冲响应（FIR）和无限长脉冲响应（IIR）两种类型讨论，如图 1.2.2 所示。

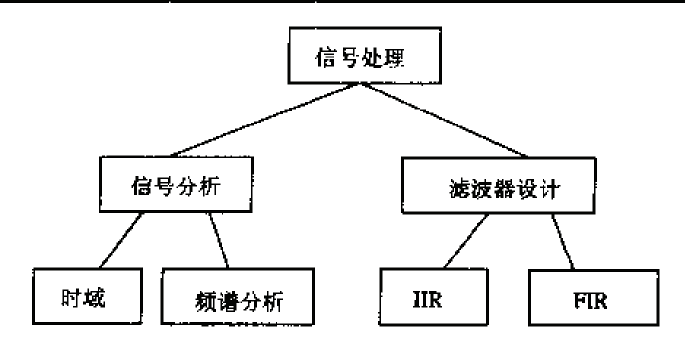


图 1.2.2 信号处理的任务示意方框图

1.2.3 数字信号处理

弱电和低频领域的信号处理任务，过去主要靠模拟电路来实现。近几十年来，已愈来愈多地依靠计算机或数字芯片来完成。这样的信号处理系统由图1.2.3组成。

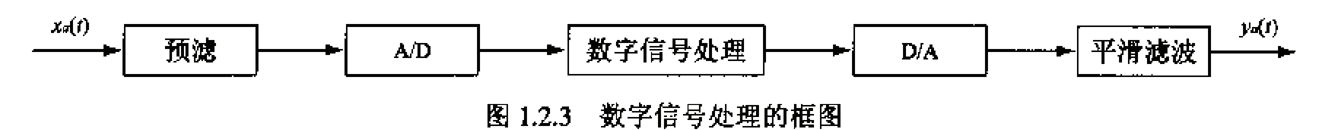


图 1.2.3 数字信号处理的框图

此系统的输入 $x_a(t)$ 和输出 $y_a(t)$ 都是模拟信号。输入信号 $x_a(t)$ 经过预滤波后进入模拟/数字(A/D)转换器，由它进行采样并量化为数字信号，然后送入数字信号处理系统进行处理。处理后的输出数字量，经过 D/A 转换器，变成模拟信号，再经过平滑滤波，成为输出信号 $y_a(t)$。送回物理系统。初看起来，这种方案增加了系统的复杂性，但实际上它却得到广泛的使用。其原因在于它比模拟信号处理具有许多优点，如下所述。

（1）灵活性

数字信号处理系统（简称数字系统）的性能取决于系统参数，这些参数保存在存储器中，很容易修改，因此系统的性能容易通过修改程序来调整。灵活性还表现在数字系统可以分时复用，用一套数字系统可以分时处理几路信号。

（2）高精度和高稳定性

模拟系统的特性会随温度、湿度等环境条件而变，数字系统的特性一般不会随使用条件变化而变化，尤其使用了超大规模集成的数字信号处理芯片(Digital Signal Processor，DSP)，设备简化，更提高了系统的稳定性和可靠性。在精度方面，模拟的繁用表只有2.5%，模拟系统达到1%就算是精密仪器了，而数字系统二进制位数可以有 8 位、16位，现在已提高到32位、64位。其精度之高是模拟系统不能相比拟的。

（3）便于开发和升级

同样的数字芯片，在通用的硬件平台上，可以开发不同软件来完成多种功能，服务于多种系统。由于软件可以方便地传送、复制和升级，系统的性能可以得到不断改善。

（4）便于大规模集成

数字部件具有高度的规范性，对电路参数要求不严，容易大规模集成和生产，这也是 DSP 芯片发展迅速的原因之一。由于采用了大规模集成电路，数字系统体积小、重量轻、可靠性和一致性高，在大规模生产的情况下，它的成本往往低于模拟系统。

（5）功能强

对数字信号可以进行存储、剪裁、复制、传送等处理，这一优点更加使数字信号处理不

再仅仅限于对模拟系统的模仿上，它可以完成许多模拟系统完不成的任务。例如，电视系统中的画中画、多画面、各种视频特技，包括画面压缩、画面放大、画面坐标旋转、动画和画面合成、特技制作等。

数字信号处理相对于模拟信号处理也有一些弱点：首先是处理速度问题。模拟系统通常是可以立即做出实时响应的物理系统，而数字信号处理则要在获得采样信号后进行规定的运算和操作，在下一个采样信号来到以前必须完成这些运算。要做到这一点，只有三个办法：一是简化运算；二是加大采样周期；三是提高运算和操作速度。正因为这些方面要受到各种技术条件的制约，目前数字信号处理还只能工作在视频以下。提高处理速度也正是数字信号处理科学和技术努力的方向。另外，涉及大功率的许多物理模拟信号处理系统，也无法由数字系统来代替。

1.3　信号采样和复原实例

利用计算机上的硬件和 Windows 操作系统，结合 MATLAB 软件，可以很方便地进行数字信号的采样、变换、存储、分析、处理和重构，现用音频信号做一个实验。

（1）在计算机的话筒输入插口上，插入一个话筒。然后打开计算机，用鼠标依次选择【开始】→【程序】→【附件】→【娱乐】→【录音机】，使录音机操作界面出现在屏幕上如图 1.3.1 所示。用鼠标单击有以红色圆点为标志的录音按钮，接着对话筒说话“大家好!”，说完后立即按下以方块为标志的停止按钮，屏幕右侧将显示所录声音的长度。例如，这里显示的是 1.25s。如果计算机上接有扬声器，所录的音就可以通过放音重现。这一步工作中，已利用了计算机上的 A/D 转换器，把模拟的声音信号，变成了离散的量化了的数字信号。而在放音时，它又通过 D/A 转换器，把保存起来的数字数据，恢复为原来的模拟的声音信号。

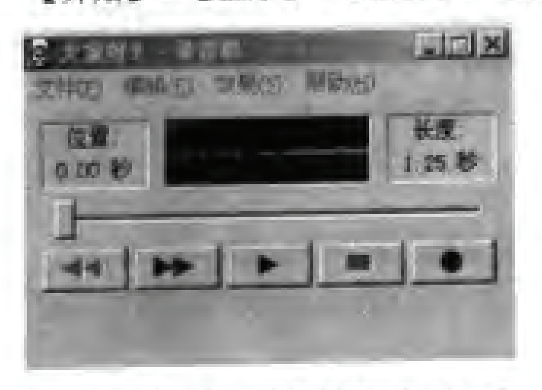


图 1.3.1　Windows 系统中的录音机界面

（2）现在做下一步的工作。单击主菜单上的【文件】→【保存】，就把这段声音以数据文件的形式存储起来。假定把文件名取为“dajiahao”(因为 MATLAB 等软件不接受中文文件名，所以用拼音)，可以看到，文件的后缀默认为.wav，这是 Windows 操作系统规定的声音文件的标准。为了进一步看出它的特征，再单击【文件】→【属性】，就可以出现如图 1.3.2 所示的文件属性界面。整个文件的数据大小为 27548B。它的采样频率是 22 050Hz，量化为 8 位二进制码，编码方式为 PCM（脉冲编码调制）。这几个数据格式是系统默认设定的。



图 1.3.2　wav 文件属性

（3）再进一步，我们还可以修改这个录音格式，也可以修改它的放音格式。先选择要转换的格式名称，再按【立即转换】，屏幕上又出现声音特性选定

界面，如图 1.3.3 所示。其中有几十项备选的声音标准，采样速率从 8 000 Hz 到 48 000 Hz 有好几种，量化等级有 8 位和 16 位两种，还有单声道和立体声的不同。读者不难从这些指标算出需要传送的数字信号的数据率。一般认为，电话是对声音质量要求最低的，只要传送 3000 Hz 以下的频率，根据采样定理，采样频率必须比所传送的频率高一倍以上，所以可以用最低的 8 kHz 采样，取单声道，量化为 8 位也够了，所以要求的传送速率仅为 8k Hz× 8 bit/s＝8 KB/s。而 CD 唱片则要求复现近 20 000 Hz 的频率，因此采样频率取到了 44 100 Hz。立体声要求传两路信号，每秒要传送 2×44 100=88 200 个采样数据，每个数据量化为 16 位，即至少要用 16 个脉冲，或两个字节来表示。可算出其要求的传送速率为 176 400 B/s，除以 1 024，得知相当于 172 KB/s。

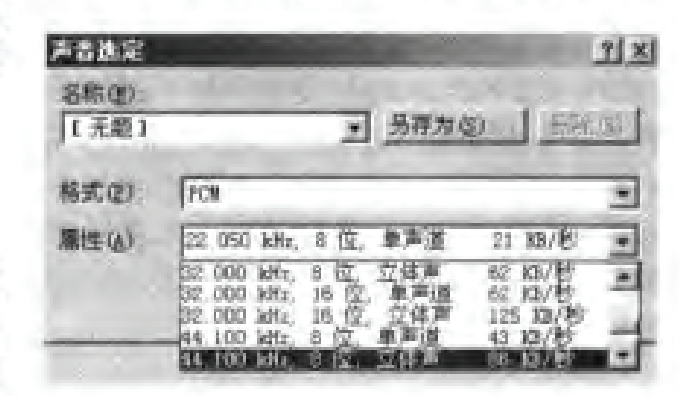


图 1.3.3　声音设定内容和文件属性

（4）再进一步，可以用 MATLAB 等软件，打开并读出文件的内容，对内容进行频谱或其他特性的分析，可以对它进行所需要的各种修改加工。首先确认数据文件 dajiahao.wav 已存放在 MATLAB 的搜索路径下，然后在命令窗中键入：

```
[x,Fs,Nbits]=wavread('dajiahao');
```

执行了这条命令，就把数据文件 dajiahao.wav 上的声音数据赋给了变量 x，同时也把 x 的采样频率 Fs 和数据的位数 Nbits 放进了 MATLAB 的工作空间。可以用 whos 命令检验。

键入 `size(x)`，

得到 `27548  1`

说明 x 为一个 27548 行的单列向量。

键入 Fs 和 Nbits 可以分别得到 22 050 Hz 和 8。如果键入：

```
x1=wavread('dajiahao',500);
```

那就是只读出声音文件上前 500 个数据，将它们赋值给变量 x1。

（5）键入 sound(x,22050)。可在计算机的扬声器中重现“大家好！”的声音。也可以观察 x 数据的具体数值。任意取 6 个数据，键入：

```
x(15120:15124)'
```

得到`-0.2266   -0.1172   -0.1172   -0.1172    0.0391    0.0391`

这是 8 位二进制数转化成的双精度数。为了证明这一点，把这组数除以 8 位二进制数的量化步长。把–1~+1 分为 2^8=256 份，量化步长为 1/128。键入：

```
x(15120:15124)'/ (1/128)
```

得到`-29   -15   -15   -15     5     5`

这是上述 6 个 x 数据的十进制数格式。它们的原始存储格式是 8 位二进制数，其中第一位是符号位，后面的 7 位是数值，x 的最小值为 -128，最大值为+127。所以 MATLAB 在读进这些数据时已经自动进行了转换。

（6）如果要想把这个声音数据改变成 MATLAB 的数据文件保存，可以用 save 命令。键入

```
save dajiahao x
```

就会自动生成一个文件名为 dajiahao.mat 的数据文件。其中以 MATLAB 的数据格式(.mat)保留了变量 x 的全部数据。这样才便于使用 MATLAB 的各种工具来处理它。而 Windows 中的声音文件 dajiahao.wav 是以二进制数格式保存的。从文件的大小就可以看出，dajiahao.mat 占 216KB，而 dajiahao.wav 只占 27KB，相差 8 倍。因为 MATLAB 中每个数都要用双精度格式保存，占用 8KB。虽然“浪费”了很多磁盘空间，但这些数据可以直接用 MATLAB 进行计算处理，节约了很多人力，又避免了人为转换可能带来的错误。这也就是设计 MATLAB 语言的基本思路。

1.4 本书概貌的说明

（1）关于对“数字信号处理”内涵的理解

对“数字信号处理”这门学科的任务可以有两种不同的理解：一种是研究对“数字信号”进行“处理”的理论和技术，这种理解把“数字信号”作为限定的研究对象，而对连续时间信号的分析计算不予考虑，这样的划分把工程中很重要的一项信号分析工作置于三不管的“盲区”；另一种则是研究如何用“数字”方法来“处理信号”，这种理解把“数字方法”作了限定，而对研究对象没有限定。那就也包括用数字方法来分析连续时间信号。我们认为第二种理解比较恰当。它比第一种含义覆盖的领域宽，消除了“盲区”。

虽然在“信号与系统”课程中，研究了连续时间信号，但所研究的信号都是能用解析形式表达的，它们的频域特性通常也都能用闭合的解析式表示。实际世界中所遇到的大部分信号是连续时间信号，很多信号是靠实测得到的，不能用解析式表示，对于这类信号进行分析，包括实时和非实时的计算，都必须借助于计算机。而计算机只能处理数量有限、数值有限的数据，也就是必须先把信号进行采样和量化，这就会带来离散信号处理中的同样问题。可见分析连续时间的模拟信号，必然要采用数字技术。现有的不少“数字信号处理”教材，就不甚关注这个问题，往往一带而过。为了克服这个缺陷，在本书第 4 章中，特别开辟了较大篇幅来讨论这个问题，以保证“数字信号处理”课程与“信号与系统”课程实现无缝的对接。

（2）关于 MATLAB 语言

既然“数字信号处理”是要研究如何用“数字方法”来处理信号的，它就应该离不开在计算机上进行实践。个人计算机的普及是条件之一，而选择适当的软件是另一个重要的条件。十多年来，由 Mathworks 公司开发的 MATLAB 软件已经成为国际信号处理界公认的数值计算和算法开发的标准平台。虽然它主要用来做非实时的计算，但也可以对实时的算法进行仿真，而且已经有 Motorola 和 TI 等公司用 MATLAB 语言作为开发其 DSP 芯片的顶层软件。这些就成为学习“数字信号处理”时，应该使用 MATLAB 语言的充分理由。

我们希望读者在学习本课之前，先学过和用过 MATLAB 语言，而不是在学习本课时边学边用。不然，读者在使用本教材的时候，会感到困难。因为本书不只把 MATLAB 作为计算工具，在讲解许多概念时，是用 MATLAB 语言来概括其思路的。

MATLAB 可以帮助学生在较短时间内，完成各种复杂的、接近于实际问题的作业，在程序中改变参数，就可以迅速地得出相应的结果。可以使读者从数字比较中加深概念的理解；

它也可以帮助教师用图形、动画来讲解复杂的概念。在本书中，自始至终应用 MATLAB 程序来计算例题。为了节省篇幅，大部分程序只给出了核心语句，完整的程序则放在程序包中。它们被放在 examples 子目录下；本书中也编了一些子程序，在运行例题程序时要调用的，放在 subrtdsp 子目录下；这两个程序包可以应读者的要求，以网上下载的方式免费提供，这部分程序是读者能够读懂的。我们还给出了一些教师的演示程序，也放在程序包中的 demos 子目录下，这些演示程序涉及一些 MATLAB 界面设计和动画设计的语句，一般的读者（包括教师和学生）并不需要掌握，只有读者想修改演示程序时才应该弄懂它们。

（3）本书的组成

本书大体上包含两部分内容：第 2～4 章是信号分析计算，主要是频谱分析和计算；第 5～8 章是系统分析设计，主要是滤波器设计。

第 1 章介绍了信号的分类，给出了各种信号处理的实例，并对信号处理的模拟方法和数字方法进行了比较。通过一个实例，告诉读者如何利用现有的个人计算机和 MATLAB 软件对模拟信号进行采集并变成数字信号并构成声音文件，还可以将它取出并重现的全部过程。使读者对数字信号有更生动实际的概念。

第 2 章讨论时域中的离散信号和系统。介绍了模拟信号及其采样序列。讨论了若干典型序列的生成方法和序列的对称性等特性，推导了序列之间的运算产生新的序列的方法，给出了卷积运算的矩阵方法。用差分方程和脉冲响应作为系统特性的时域表述，讨论了如何用脉冲响应分析系统的特性。

第 3 章讨论频域中的离散信号和系统。讨论了数字频率、序列频谱以及系统频率特性的概念，介绍了计算序列频谱的理论基础：详细说明了离散时间傅里叶变换(DTFT)和离散傅里叶变换(DFT)的特性。特别研究了 DFT 的循环特性，比较了循环卷积与线性卷积的关系，讨论了用离散傅里叶变换进行分段卷积的计算方法。

第 4 章讨论各类信号的频谱分析和计算的高效方法。首先介绍快速计算傅里叶变换（FFT）的方法。然后依次讨论如何用 FFT 来计算有限序列、无限序列和模拟信号频谱的问题；也讨论了逆问题，即根据已知频谱由傅里叶反变换(IFFT)求时间序列或模拟信号的问题。讨论了造成误差的原因和影响因素，研究了提高计算精度和速度的措施。这一章的内容在国内现有教材中是较少涉及的，大部分内容来自参考文献[3]。

第 5 章讨论离散线性时不变集总参数系统的 z 变换域表示方法。首先研究了 z 变换及其收敛性，讨论了 z 反变换的多种算法。把它在时域分析中的时移特性与 z 域的复频率特性统一起来。根据 z 变换作为傅里叶变换在复平面上的推广，讨论了零极点分布对系统频率特性的影响。介绍了几种零极点分布比较特殊的滤波器。

第 6 章讨论数字滤波器结构。采用不同的运算结构可以实现相同的滤波器传递函数，但在考虑到量化效应的情况下，其精度和速度可能很不相同。这是实现滤波器时的重要问题。本章介绍了用信号流图描述线性系统的一般方程组及其求解公式，利用时移算子 z 把信号流图和状态方程联系起来，与 MATLAB 的符号推理工具箱相结合，得到求任何复杂滤波器传递函数的一般方法，这个方法是作者在参考文献[14]中创立的。本章还介绍了滤波器的几种常用结构形式，讨论了它们之间系数转换的关系式，以及相应的 MATLAB 函数。

第 7 章研究 FIR 滤波器的设计。首先介绍四种线性相位滤波器的特征以及它们的适用场合，以后着重讨论三种方法：窗函数法、频率采样法和最优化方法。鉴于在第 4 章中对序列及其频谱所做的深入讨论和实际计算，读者掌握本章将比较容易。另外，充分利用信号处理

工具箱函数进行设计也将是本书与其他国内教材的不同之处。

第 8 章研究 IIR 滤波器的设计。先讨论模拟原型滤波器的设计方法，包括模拟域内变换为高通、带通等滤波器的函数，这样读者可以完整地掌握模拟滤波器的设计技术，而且信号处理工具箱函数已经有这些函数可供读者调用。然后再讲数字与模拟频率转换的脉冲响应不变和双线性两种方法。后面也讲数字域内低通到带通等转换方法。本章还介绍了 MATLAB 中的滤波器设计工具和信号处理工具，为读者提供更广的思路。

第 9 章给出了综合应用数字信号处理技术，解决实际问题的若干例子。首先介绍了 MATLAB 提供的在时域和频域对信号进行分析的环境 sptool。在这个环境中可以对信号、滤波器及频谱进行统一命名和管理。可以让它们相互变换、作用，也可以进行观测和修改。这样一个综合环境相当于对课程起了归纳作用，并且使读者能很快地把理论用于工程。接着又举了七个应用实例：PCM 语音传送中的 A/D 转换问题；系数和运算量化影响的分析方法；双音拨号系统中的数字处理问题；频谱分析仪中的算法问题；音乐信号处理技术；变采样率滤波器设计问题和稀疏天线阵的设计问题等。这些问题都是以前八章中的理论为基础，同时也都需要做一些引申和扩展。有些还可以变成教学实验题目。最后简单介绍了数字信号处理的发展和 MATLAB 中可以提供的工具。

复习思考题

1.1 时域离散信号和数字信号有什么不同？

1.2 在数字电子计算机上处理的信号是模拟信号、时域离散信号还是数字信号？

1.3 为什么在传输和处理中，需把数字信号编成二进制码？

1.4 数字技术比模拟技术有哪些优点？

1.5 什么是幅度量化？什么是时间量化？

1.6 什么是量化步长？如何求出给定数（或数组）的量化值？

习题

1.1 设一组电压值为 x=[−0.3，0.5，1，−2，3，−4]，经过一个量化步长为 0.15 的量化装置，求量化后的整数和它们对应的量化电压。求出其绝对误差和相对误差。

1.2 同上题的输入电压，如果量化器是一个最大输入幅度±5V，8 位（含符号位）的 A/D 变换器，求出其绝对误差和相对误差。

1.3 一组电压值为 x=[0:0.5:4]，经过一个把−5～5V 正电压转换为 12 位（包括符号位）二进制的 A/D 转换器，求输出的量化电压的二进制代码，并求经 D/A 转换后的量化电压值。

1.4 一个图形分为 525 行，每一行有 650 点，每个点用 24 位二进制编码，则描述这个图形共要多少个二进制位？如果每秒要传送 25 帧图形，则传送速率为多少（单位为 bit / s）？

1.5 设数字信号的采样周期为 0.1 ms，对它进行处理需要四次乘法和二次加法。所采用的信号处理器乘法所需的时间为 10 ns（10×10^{-9}s），加法所需的时间为 0.5 ns（0.5×10^{-9}s），问用这样的信号处理器能同时处理多少路这样的数字信号？

第2章　时域中的离散信号和系统

2.1　模拟信号与其采样序列的关系

对模拟信号 $x_a(t)$ 以采样周期 T 进行等间隔采样，得到

$$x(n) = x_a(t)\big|_{t=nT} = x_a(nT) \qquad -\infty < n < \infty \tag{2.1.1}$$

这里，n 取顺序增加的整数；$x_a(nT)$ 是按时间先后排列的离散序列。在实际信号处理中，这些数字序列值 $x(n)$按顺序放在存储器中，此时 n 代表的是前后顺序，该数字序列 $x(n)$称为时域离散信号，简称序列。很多书在叙述文中用$\{x[n]\}$来称呼一个序列，由于 MATLAB 中 x 就可表示数组或序列，且其下标用圆括号。以后我们将逐步淡化这个区别，在可能的情况下，将尽量简单地用 MATLAB 的符号规则同时完成叙述和编程两项工作。

离散序列 $x(n)$代表了模拟信号 $x_a(t)$ 在一些等间隔的离散时刻上的值。但它在物理上已经不能与原模拟信号等价了，因为它只有一个幅度，作用时间是无穷小，没有了作用的时间。从力学的概念说，它没有了动量（力乘以作用时间）和能量（力乘以作用距离）；从电学上看，这些幅度有限又没有作用时间的信号序列，连推动数字电路翻转的能力都不具备，更不要说推动扬声器放音了。

要使数字信号能与对应的模拟信号有相似的物理表现，应该是把模拟信号在采样时刻 nT 前后各 $0.5T$ 的面积（相当于力学中的动量）集中到采样时刻 nT 上，其面积为 $T \cdot x(nT)$，这样的离散信号才能与对应的模拟信号在物理上大体等价。要表达这样的信号，应当用 δ 函数，它具有无限窄的时间，无限大的幅度，而所围的面积则是有限的。从这个角度出发，在物理上和数学上与原模拟信号相当（或近似等价）的采样信号应该是：

$$x_e(n) = T \cdot x(n) = T \cdot x_a(nT) \tag{2.1.2}$$

我们说离散信号 $x_e(n)$ 可以作为与原模拟信号 $x_a(t)$ 近似等价，就是因为当 $T \to 0$ 时，$x_e(n)$ 与 t 轴的任意区间所围的面积与 $x_a(t)$ 与该 t 轴区间所围的面积相同。

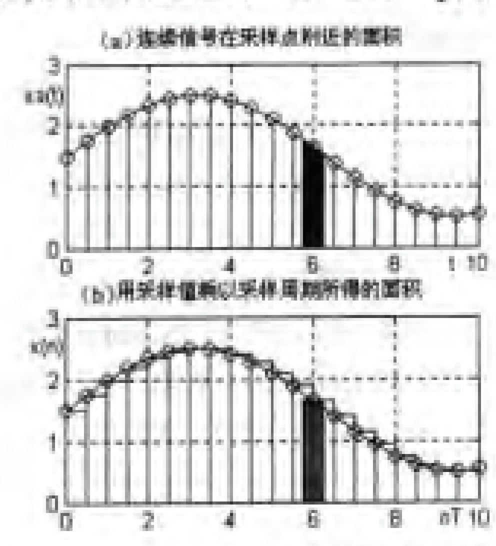


图 2.1.1　连续信号与采样信号的面积等价

把采样序列 $x(n)$ 和等价序列 $x_e(n)$ 的定义加以比较，可以看出它们在量纲上就差了一个时间，因此，采样序列在物理上并不等价于原信号。比如，在把数字信号的频谱与原模拟信号频谱作比较时，要注意两者之间差了一个乘子 T；当想把数字序列复原为模拟信号时，也必须乘以采样周期 T，采用的 D/A 转换器通常都把转换出的电平保持一个采样周期，直到下一个数字信号到达。其输出信号相当于图 2.1.1（b）图中的阶梯形，也相当于把它乘以 T。

实际上，为了使数字信号处理的理论完全建立在

序列运算的基础上，要使它更高地抽象，不受采样周期 T 的影响。人们用（2.1.1）式得出的采样序列作为离散信号的研究模型，它不涉及真正的时间和物理量。模型越抽象，它就越具有更大的普遍意义。在数字域处理信号的整个过程都可以不管 T，只对序列进行运算，本书中大部分地方均采用这种方法。只有在与模拟域信号进行转换和比较的时候，才需要按(2.1.1)式来考虑 T 及与模拟信号等价的问题。

2.2 基本序列及其运算

2.2.1 序列的分类表示方法

当把一个经等间隔采样的信号表示为序列后，就抽掉了它的采样周期 T 的信息。如果序列的长度有限，就可以用一个向量来表示。然而这样一个向量并没有包含采样位置（或时间先后）的信息。因此，完全地表示序列 $x(n)$ 要用 x 和 n 两个向量，例如序列 $x(n)=\{2,1.2,-1.4,\underline{\underline{3}},1,4,3.1,7\}$（下划双线表示 n=0 处的采样点），即 $x(-3)=2$, $x(-2)=1.2$, $x(-1)=-1.4$, $x(0)=3$, …，所以要用 n 和 x 两个向量才能完全地表示一个序列：

$n = [-3,-2,-1,0,1,2,3,4];$

$x = [2,1.2,-1.4,3,1,4,3.1,7];$

位置向量 n 是顺序增加的整数，所以在 MATLAB 中可以简化表示为 n=[-3:4]；而因变量 x 是一个模拟量数组，可以有小数，甚至可以是无理数。有些序列，因为采样位置从 n=0 开始，就省写了 n 向量，只用 x 向量来表示序列 $x[n]$。在 MATLAB 的信号处理工具箱函数中，对输入序列大多都采取这种设定，以减少输入变元的数目，实际上 n 仍然存在。如果 x 的长度是 N= length(x)，即默认 n=0:N-1。由于 MATLAB 中的变量下标从 1 开始，此时 x 的下标仍不是 n，而应该是 n+1，x 的下标范围为(1:N)。所以即使是 MATLAB 的默认条件下，x 的下标并不等于位置向量 n。本书中给出的序列运算的函数程序，输入和输出序列都用 n 和 x 两个向量表示。所以存在一个与 MATLAB 的信号处理工具箱函数的衔接问题，需要做简单的处理。

如果 x 是实数数组，则称为实序列；如果 x 是复数数组，则称为复序列；一个复序列可以分解为两个实序列的合成：

$$x(n) = x_{\text{re}}(n) + \text{j}x_{\text{im}}(n) \tag{2.2.1}$$

其中 $x_{\text{re}}(n)$和$\text{j}x_{\text{im}}(n)$ 都是实序列，它们分别为序列 $x(n)$的实部和虚部。因此可以用实序列的理论和方法来处理。复序列也可以用它的幅度和相角来表示：

$$x(n) = |x(n)|\text{e}^{\text{j}\varphi(n)} \tag{2.2.2}$$

其中幅度$|x(n)|$和相角 $\varphi(n)$也都是实序列。

如果 x 的长度 N= length(x)有界，称之为有限序列；如果 N 无界，称之为无限长序列。实际中遇到的序列都是有限序列，无限长序列是一种数学模型，只有理论意义。在无限长序列中，若 n 向+∞延伸时 $x(n)\neq 0$，称为右序列；若 n 向-∞延伸时 $x(n)\neq 0$，称为左序列。若 n 向-∞和+∞双向延伸时，都有 $x(n)\neq 0$，称为双向无限序列。

2.2.2　常用的典型序列

（1）单位脉冲序列 $\delta(n)$

$$\delta(n)=\begin{cases}1 & n=0\\0 & 其余\end{cases} \tag{2.2.3}$$

其特点是仅在 n=0 处取值为 1 ，n 为其他值处均为零。它类似于模拟信号中的单位冲激函数 $\delta(t)$。

后面常常要用到迟延的单位脉冲序列。其脉冲的作用起点不是 0 而是 n_p，表为：

$$x_1(n)=\delta(n-n_p)=\begin{cases}1 & n=n_p\\0 & 其余\end{cases} \tag{2.2.4}$$

为了调用的方便，本书建立一个MATLAB函数impseq.m，用来生成迟延的单位脉冲序列。其输入参数为序列起始位置ns, 序列终止位置nf及脉冲位置np。函数如下：

```
function [x,n] = impseq(np,ns,nf)
% 生成 x(n) = delta(n-np); ns <= n <= nf
% ---------------------------------------------
% ns 为序列起始位置, nf 为序列终止位置, np 为脉冲位置
% 调用方式  [x,n] = impseq(ns,ns,nf)
%
if ns>np|ns>nf|np>nf
error ('输入位置参数不满足ns <= np <= nf')
else n = [ns:nf];
x = [(n-np) == 0];
end % 单位脉冲序列的产生
```

这里出现了本书第一个自编的 MATLAB 函数文件，写得比较完整。程序中的核心部分是第一行和倒数第二行，缺了它们程序不能工作。第二至六行都以%开始，它们是注释语句。在键入 help impseq 时将显示这几行内容，以便用户正确调用这个函数。以 if 打头的第七、八两行是用来检验输入参数正确性的。输入参数错误时它会显示错误的性质。从 MATLAB 语法上用了完整的 if-else-end 语句。今后本书中的函数程序，将省略掉某些辅助语句，以节省篇幅。但在我们提供的程序包中，将给出完整的程序。

注意倒数第二行的核心语句采用了逻辑语句 `x = [(n-np) == 0]`。在 n=ns:nf 的一串值中，只有一个值能满足这个逻辑式，因而只在这个 n=np 处，x 为 1，其余的 n 值处，x 均为零,形成了迟延 np 的单位脉冲序列 x。读者也可以试用其他方法来写这条核心语句，不过本书提供的这个语句可能是最简练的。

（2）单位阶跃序列 $\mu(n)$

$$\mu(n)=\begin{cases}1 & n\geq 0\\0 & n<0\end{cases} \tag{2.2.5}$$

它类似于模拟信号这的单位阶跃函数 $\mu(t)$, $\mu(n)$和 $\delta(n)$的关系如下

$$\delta(n)=\mu(n)-\mu(n-1) \tag{2.2.6}$$

$$\mu(n)=\sum_{k=0}^{\infty}\delta(n-k)=\sum_{m=-\infty}^{n}\delta(m) \tag{2.2.7}$$

由于后面常会用到这个序列。本书也建立一个函数 stepseq.m，用来生成迟延的单位阶跃序列。其输入参数为序列起始位置 ns, 序列终止位置 nf 及阶跃位置 np。函数如下

```
function [x,n] = stepseq(np,ns,nf)
n = [ns:nf]; x = [(n-np) >= 0]; % 单位阶跃序列的产生
```

它与 impseq 函数的差别只是把逻辑语句中的“等于”改成了“大于等于”，读者也可以尝试用其他语句来实现此程序中的核心语句。

（3） 矩形序列 $R_N(n)$

$$R_N(n)=\begin{cases}1 & 0\leqslant n\leqslant N-1\\ 0 & \text{其他}n\end{cases} \tag{2.2.8}$$

式中的 N 为序列的长度。矩形序列可以方便地用单位阶跃序列之差来生成：

$$R_N(n)=\mu(n)-\mu(n-N) \tag{2.2.9}$$

所以不必为它编了程序了。

（4）复指数序列

$$x_3(n)=\begin{cases}e^{(\alpha+j\omega)n} & n\geqslant 0\\ 0 & n<0\end{cases} \tag{2.2.10}$$

当 ω=0 时，它是实指数序列；如 α=0，它是虚指数序列 $e^{j\omega n}=\cos(\omega n)+j\sin(\omega n)$，故其实部为余弦序列，虚部为正弦序列。

例 2.2.1：编写 MATLAB 程序来产生下列基本脉冲序列。

a.单位脉冲序列：起点 n_s，终点 n_f，在 n_p=3 处有一单位脉冲($n_s\leqslant n_p\leqslant n_f$)，

b.单位阶跃序列：起点 n_s，终点 n_f，在 n_p=3 前为 0,在 n_s 处及以后为 1($n_s\leqslant n_p\leqslant n_f$)，

c. 复数指数序列：取 α= - 0.2，ω=0.5，起点为 n_{s3} = - 2，终点 n_f。

解：这些基本序列的表达式比较简单，编写程序也不难。对单位脉冲序列和单位阶跃序列，我们直接调用了上面编的子程序。编写的程序 hc221 如下：

```
clear,ns=0;nf=10;np=3;ns3=-2
[x1,n1]=impseq(np,ns,nf); % 单位脉冲序列的产生
[x2,n2]=stepseq(np,ns,nf); %单位阶跃序列的产生
n3 = ns3:nf; x3=exp((-0.2+0.5j)*n3); % 复数指数序列
subplot(2,2,1),stem(n1,x1);title('单位脉冲序列')
subplot(2,2,3),stem(n2,x2,'.');title('单位阶跃序列')
subplot(2,2,2),stem(n3,real(x3),'x');line([-5,10],[0,0]) %画横坐标
title('复指数序列'),ylabel('实部')
subplot(2,2,4),stem(n3,imag(x3),'filled');
line([-5,10],[0,0]),ylabel('虚部')
```

程序运行结果如图 2.2.1 所示。

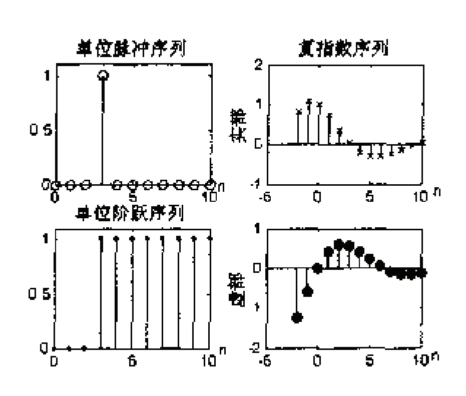


图 2.2.1　几种基本序列的波形图

在程序中采用了 stem 绘图函数来代替 plot。因为 plot 是用来画连续曲线的，它自动把各点用直线连接起来，所以不适于表现脉冲序列。本例中几条绘图语句 stem 使用了不同的“点形”输入参数，因此各条曲线上的标志方式不同。读者可以参照使用。因为 stem 命令是绘制序列时用得很频繁的，要熟悉它的用法。在后面的程序中，将省略大同小异的 subplot,xlabel,title, 以及 plot,stem,line 等绘图和标注语句。

（5）正余弦序列

$$x(n)=\cos(\Omega_0 n+\theta),\quad -\infty<n<\infty$$

其中 θ 为以弧度为单位的相角，例如，要产生：

$$x(n)=3\cos(0.1\pi n+\pi/3)+2\sin(0.5\pi n),\ 0\leqslant n\leqslant 10$$

可以用下列 MATLAB 语句：

```
n = [0:10]; x = 3*cos(0.1*pi*n+pi/3) + 2*sin(0.5*pi*n);
```

（6）随机序列

许多实际的序列是不能用确定性的数学式来描述的。这些序列称为随机序列。在 MATLAB 中,有两种基本的人造的随机序列（也称伪随机序列）可供调用。rand(*N*,*M*)函数产生在[0,1]之间均匀分布的 $N\times M$ 元随机矩阵。randn(*N*, *M*)则产生均值为 0,方差为 1 的 $N\times M$ 元正态分布高斯随机序列。

2.2.3　序列的运算和变换

广义地说，对序列的运算包括了对时域序列和频域序列的运算。通常有以下的几种运算：乘法、加法、倍乘、移位、折叠、卷积、相关等。频域序列是复数，它还有求共扼、求模或相角等复数的运算。与复数序列运算有关的深入问题将在以后讨论。序列的基本运算类型见图2.2.2。其中(e)是(d)的逆运算，而(f)是分叉，不属于运算。因此主要是(a)、(b)、(c)、(d)四种，即乘法、加法、倍乘和移位。

下面将简要地介绍序列的基本运算和实现它们的 MATLAB 程序。

（1）序列的乘法［图 2.2.2 中（a)］：$y(n)=x1(n)\times x2(n)$

序列的乘法是一种非线性运算，它用于信号的调制。它把两个序列中位置序号相同的样本相乘，形成新的样本序列。例如：$y(n)=x(n)\sin(\omega_0 n+0.1\pi)$称做对 $x(n)$做正弦调制；$y(n)=x(n)\cdot R_N(n)$为从 x 序列中用矩形函数 $R_N(n)$ 截取[0，N-1]的一段有限序列。

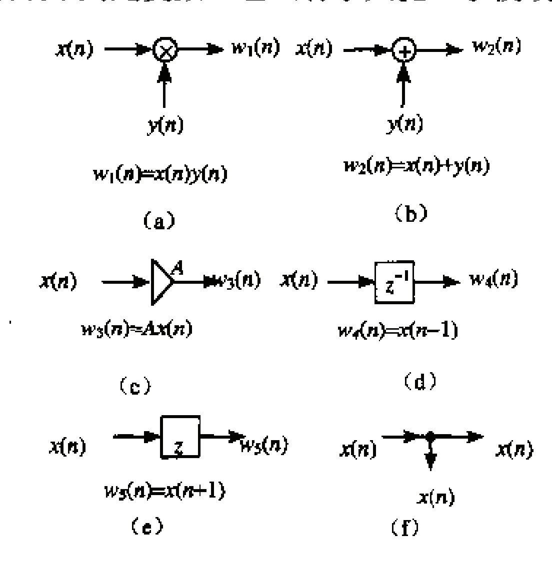


图 2.2.2　序列运算的基本类型

（2）序列的加法［图 2.2.2 中（b)］：$y(n)=x1(n)+x2(n)$

加法用于序列的合成。它把两个序列中位置序号相同的样本相加，形成新的样本序列。

在 MATLAB 中序列加法可用算术运算符“+”实现，序列乘法要用点乘运算符“.*”实现。然而首先必须要使 *x*1 和 *x*2 的位置向量的起点、终点相同，因而长度相等。如果序列长度不等，或者长度虽然相等但采样的位置不同，就不能用运算符“+”和“.*”了。

下面来编写实现这两种运算的子程序。首先要使 *x*1 和 *x*2 有同样的位置向量 n。这需要非常注意 MATLAB 的下标运算,先要把 *x*1 和 *x*2 的位置向量 *n*1 和 *n*2 扩展成包括两者最左和最右端点的公共位置向量 *n*，显然 min(*n*)应该是 min(*n*1)和 min(*n*2)中的小者，而 max(*n*) 应该是 max(*n*1)和 max(*n*2)中的大者。然后把 *x*1 和 *x*2 放到扩展后的位置向量 *n* 上，因为 *n* 可能比 *n*1 和 *n*2 长，必须把 *x*1 和 *x*2 的长度加大，把加长后的 *x*1 和 *x*2 分别命名为 *y*1 和 *y*2。这需要

把 x1 和 x2 放到位置向量 n 的对应的位置上，并把多余的元素置零。

程序中先把 y1 和 y2 全置零，然后用 find 函数来寻找对应的下标区间。因为 find 函数的输出是满足条件的输入变量的下标值。当输入条件指定了 n 中与 $n1$(或 $n2$)相重合的部分，则 find 函数输出的是重合的部分的下标。把 x1 和 x2 分别放到这个位置上就构成了 y1 和 y2。设第一个序列用 x1,n1 两个向量表示，第二个序列用 x2,n2 两个向量表示，输出序列用 y,n 两个向量表示。则下面的 MATLAB 程序将完成这个运算：

```
n = min(min(n1),min(n2)):max(max(n1),max(n2)); % 确定位置向量n
y1 = zeros(1,length(n)); y2 = y1;             % 把y 向量初始化为全零
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;    % 把x1 延拓到n上生成y1
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;    % 把x2 延拓到n上生成y2
y = y1+y2;                                     % 把两序列相加（相乘时改为y=y1.*y2 ）
```

可以把这个程序写成函数形式。命名为 seqadd,在这五行程序之前，加上函数程序头语句：

```
function [y,n]=seqadd(x1,n1,x2,n2)
```

序列相乘的函数命名为 seqmult, 其函数程序头语句为：

```
function [y,n]=seqmult(x1,n1,x2,n2)
```

其程序内容的前四行与 seqadd 相同，只需把第五行换成 y=y1.*y2；即可。

从这两个子程序可以看出，在许多情况下，序列的位置向量对序列的运算起着不可忽视的作用。在 MATLAB 的信号处理工具箱中，没有序列运算的函数，所以没有强调位置向量。在有些情况下，如不注意，容易造成错误。

（3）倍乘[图 2.2.2(c)]: 在此运算中，把序列中所有的元素乘以一个常数 a，得：

$$y(n)= a\cdot x(n) \tag{2.2.11}$$

在 MATLAB 中可用乘法运算符来实现倍率运算，即 y=alpha*x。要注意倍乘和乘法的不同：倍乘是乘以常数，是线性运算；乘法运算指序列乘以序列，是非线性运算。

（4）移位[图 2.2.2(d)]：在这个运算中，$x[n]$的每一个样本都向右移动（即迟延）k 个采样周期,设移位后的序列为 $y(n)$。

$$y(n)=x(n-k) \tag{2.2.12}$$

在 MATLAB 中，如果原始的序列用 x 和 nx 表示，移位后的序列用 y 和 ny 表示，移位运算并不影响向量 x 的值，因此 $y=x$。移位体现为位置向量的改变。ny 的每个元素都比 nx 加了一个 k，即 $ny=nx+k$。y 和 ny 就是移位后向量的表述，说明 y 取 k 拍前的 x 值。向左移位可令 k 取负号，意味着 y 取 k 拍后的 x 值。把移位函数命名为 seqshift，可以写出：

```
function [y,ny]=seqshift(x,nx,k)
y=x; ny=nx+k;
```

在图 2.2.2（d）中，方框中用 z^{-1} 进行标注，它被称为迟延算子，表示把输入序列右移一位（拍)；而在（e）中，方框中用 z 进行标注，它是左移算子，是右移算子的逆运算。实际上，迟延算子取的是序列过去的值，具有物理可实现性；而左移算子是提前算子，它要知道序列未来的值，物理上无法直接实现。所以数字信号处理中通常都用 z^{-1} 算子。

（5）折叠：在这个运算中，$x(n)$的每个样本都对纵坐标轴 $n=0$ 做对称翻转(flip-flop)，得到一个折叠后的序列 $y(n)$。

$$y(n)=x(-n) \tag{2.2.13}$$

在 MATLAB 中，这一运算由两次调用 fliplr 函数来实现。y=fliplr(x)的作用是把行向量 x 中元素排列的次序左右翻转。ny=-fliplr(nx) 既将其位置向量左右翻转，又变了正负号。相当于对 n=0 点左右翻转。折叠的结果不仅是输出的排列次序反了，而且把 $n>0$ 的位置倒写为 $n<0$。所以这两条语句就决定了折叠后的向量 y 和 ny，写成一个折叠函数 seqfold.m, 该子程序具有以下形式：

```
function [y,ny]=seqfold(x,nx)
y=fliplr(x); ny=-fliplr(nx);
```

（6）序列能量：复数序列 $x(n)$ 的能量由下式给出：

$$E_x=\sum_{n-\infty}^{\infty}|x(n)|^2=\sum_{n-\infty}^{\infty}x(n)x^*(n) \tag{2.2.14}$$

其中上标 * 表示共轭运算。有限长度序列 x 的能量可用以下的 MATLAB 命令求得：

```
Ex = sum(x .* conj(x));       % 一种方法
Ex = sum(abs(x) .^ 2);        % 另一种方法
```

（7）序列功率：基本周期为 N 的周期序列的平均功率由下式求得：

$$P_x=\frac{1}{N}\sum_{n=0}^{N-1}|x(n)|^2 \tag{2.2.15}$$

实现这一运算的 MATLAB 命令与求信号能量相仿，只是求 N 项的和后，再除以 N。

前面已经指出，序列的幅度与模拟信号幅度物理上不等价，所以序列的能量和功率也只是离散域中与连续域能量和功率计算形式相仿的运算，并不等同于物理上的能量和功率。

例 2.2.2：画出以下各序列在给定区间的波形图。

a. $x(n)=1.5*\delta(n+1)-\delta(n-3)\quad -4\leqslant n\leqslant 5$

b. $x(n)=n[\mu(n)-\mu(n-8)]-10e^{-0.3(n-10)}[\mu(n-10)-\mu(n-16)]\quad 0\leqslant n\leqslant 20$

c. $x(n)=\cos(0.07\pi n)+0.2w(n)\quad 0\leqslant n\leqslant 30$ 其中 $w(n)$为具有零均值的单位方差的高斯随机序列。

d. b 中函数 $x(n)$的能量 $E(n)$

解：在做这些例题时，建议读者先根据概念在草稿纸上用笔草算和画图，再用程序验证。程序 hc222 的核心语句如下：

```
n1=[-4:5]; x1=1.5*impseq(-1,-4,5)-impseq(3,-4,5);          % 对题a.
n2=[0:20];x21 = n2.*(stepseq(0,0,20)-stepseq(8,0,20));     % 对题b.
x22 = 10*exp(-0.3*(n2-10)).*(stepseq(10,0,20)-stepseq(16,0,20));
x2 = x21-x22;
n3 = [0:30]; x3 = cos(0.07*pi*n3)+0.2*randn(size(n3));     % 对题c.
n4=n2; E=x2.^2;                                             % 对题d.
```

在四个子图中分别用 `stem(n1,x1)`,…, `stem(n4,E)`绘图，可以得到图 2.2.2。注意能量图永远取正数。

在本例中，由于所有参与运算的序列的位置向量都相同，所以可直接用加号或点乘完成序列运算，程序相对简单。下面的例子将考虑序列位置向量不同的情况，以便说明子程序 seqadd 和 seqmult 的运算过程。

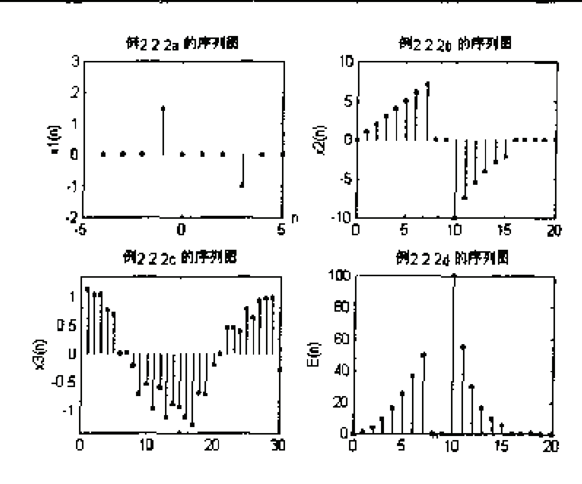


图 2.2.2　序列的简单运算与合成

例 2.2.3：给出两个序列 $x1$ 和 $x2$。

$x1$=[0 ,1, 2 ,3 ,4 ,3 ,2 ,1 ,0]; 起始位置 ns1=-2;

$x2$=[2,2,0,0,0,-2,-2], 起始位置 ns2=2;

求它们的和 ya 及乘积 yp，并画出序列 $x1$，$x2$，中间序列 y1，y2 及 ya，yp 图，并对这些图作出解释。

解：在编写解题程序 hc223 时，因为要画出中间出现的序列 y1，y2，所以不调用子程序，直接在主程序放进前述运算语句。

```
x1=[0,1,2,3,4,3,2,1,0];ns1=-2;                    % 给定x1 及ns1
x2=[2,2,0,0,0,-2,-2]; ns2=2;                      % 给定x2 及ns2
nf1=ns1+length(x1)-1;                             % 求出x1，x2 的终点位置nf1,nf2
nf2=ns2+length(x2)-1;                             % 由此得知nx1=ns1:nf1,nx2=ns2:nf2
ny= min(ns1,ns2):max(nf1,nf2);                    % y 的位置向量
y1 = zeros(1,length(ny)); y2 = y1;                % y1,y2 序列初始化
y1(find((ny>=ns1)&(ny<=nf1)==1))=x1;              % 给y1 赋值x1
y2(find((ny>=ns2)&(ny<=nf2)==1))=x2;              %  给y2 赋值x2
ya = y1 + y2                                      % 序列相加
yp = y1.* y2                                      % 序列相乘
```

作图语句为 stem(nx1,x1)等，把输入 $x1$,$x2$ 放在左边，右边则安排 y1，y2，ya，yp。因为它们有公共的位置向量 ny。结果如图 2.2.3 所示。

注意位置向量 ny=-2:8，它覆盖了位置向量 $nx1$=-2:6 及 $nx2$=2:8 的全部。y1 和 y2 就是在加长了的公共位置向量 ny 上的 $x1$ 和 $x2$。因此 y1 和 y2 才能直接相加和相乘。

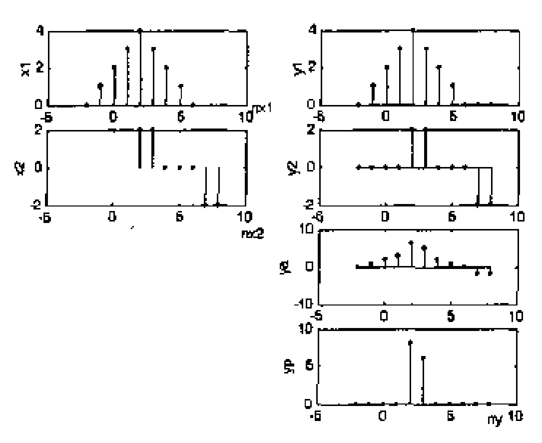


图 2.2.3　位置向量不同的两序列的相加和相乘

2.3　几种重要的序列和序列运算

复杂的序列通常可由基本序列通过运算和组合构成的,下面将举出几个在数字信号处理中常遇到的重要序列及运算。

2.3.1　周期序列

如果对所有的 n，序列 $\tilde{x}[n]$ 满足

$$\tilde{x}[n] = \tilde{x}[n+kN] \tag{2.3.1}$$

则 $\tilde{x}[n]$ 称为周期序列。满足上述关系的最小数 N 称为基本周期。

用 MATLAB 把一个周期序列 $\{x(n), 0 \leqslant n \leqslant N-1\}$，拓展为有 K 个周期的序列 xtide (n)，有多种方法可以采用，这里介绍两种。

（1）简单复制法：设 x 是一个已赋值的行向量，其长度为 N=length(x),把它复制 K 次，得到的 xtide 长度为 $K*N$。因此有：

```
xtide = [X,X,...,X]; nxtide=0:K*N-1
```

这个方法在 K 太大时容易数错，不是很好。

（2）求余函数 mod 法：函数 $n1$=mod(n,N)完成运算 n1=(n mod N)。这个算式把大于等于 N 的 n 值，减去 N 的整倍数，使余数 $n1$ 在 0 与 $N-1$ 之间。对于小于等于 0 的 n 值，则加以 N 的整倍数。也使 $n1$ 在 0 与 $N-1$ 之间。把这一运算用到位置向量上，就可以方便地实现有限序列的周期延拓。

先设置位置向量，要复制 K 个 x，则新向量的长度应为 $K*N$。其中 N 为 x 的长度，也就是所取的周期。假如起始位置为 0，则可用下列语句：

```
nxtide=0:K*N-1;                    % 设置延拓序列的位置向量
xtide=x(mod(nxtide,N)+1)           % 确定位置向量各点对应的x值
```

在最后一条语句中，注意 mod(nxtide,N)的值永远在 0 与 $N-1$ 之间，而按 MATLAB 的规则，x 的下标标为 nx=1:N。因此必须把 mod 函数的结果加 1 才能与下标规则匹配，并把所有的 x 都依次取到。可见随着 nxtide 中序号的不断增加，x 的下标将周期性地改变，xtide 就会周期性地取 x 的值。例如若 N=7 而 ntide=16 时，xtide=$x(16-2\times7+1)=x(2)$。

例 2.3.1：设 x=[2,3,4,5]，求将它延拓 6 个周期所得的序列。

解：本题的 MATLAB 程序 hc231 如下

```
x =[2,3,4,5]; N=length(x); K=6;
ny=0:K*N-1;            % 设置延拓序列的位置向量
y=x(mod(ny,N) + 1)  % 确定位置向量各点对应的x值
```

用 stem(nx,x)及 stem(ny,y)绘图，得出的图形见图 2.3.1。

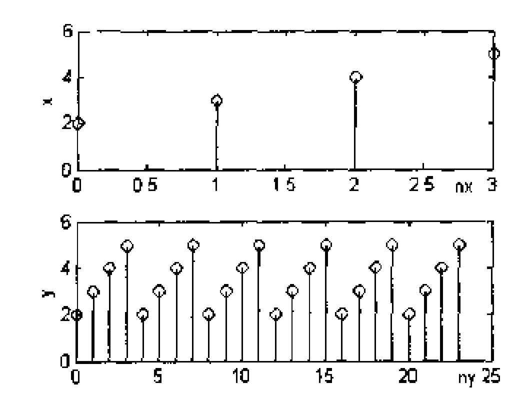


图 2.3.1　序列的周期延拓

2.3.2　对称序列

一个实数序列 $x_e(n)$，如果满足 $x_e(-n)=x_e(n)$，就称为偶序列，或对称序列。反之，一个

实数序列 $x_o(n)$ 如果满足 $x_o(-n)=-x_o(n)$，就称为奇序列，或反对称序列。

定理：任何任意实数序列 $x(n)$都可以分解成偶序列和奇序列两个分量，见式（2.3.2）。

$$x(n)= x_e(n)+ x_o(n) \tag{2.3.2}$$

其中偶部和奇部分别由（2.3.3a）和（2.3.3b）式给出。

$$x_o(n)= [x(n)+x(-n)]/2 \tag{2.3.3a}$$

$$x_o(n)= [x(n)-x(-n)]/2 \tag{2.3.3b}$$

这个定理是不难证明的。只要证明，按(2.3.2)式算出的 $x_o(n)$和 $x_o(n)$满足(2.3.2)式，同时，它们又分别满足偶序列和奇序列的定义 $x_e(-n)= x_e(n)$和 $x_o(-n)=-x_o(n)$即可。这个证明留给读者自行完成。

在研究傅里叶变换的特性时会用到这个定理。利用前面建立的序列基本运算的函数，不难得到把任意实序列 x 分解为偶序列 xe 和奇序列 xo 的 MATLAB 语句如下：

```
[x1,nx1]=seqfold(x,nx);                   % 求出序列 x1(n)=x(-n)
[xe,nxe]=seqadd(x/2,nx,x1/2,nx1)          % 按(2.3.3)求出序列 xe
[xo,nxo]=seqadd(x/2,nx,-x1/2,nx1)         % 按(2.3.3)求出序列 xo
```

其实，位置向量 nxe 和 nxo 是相同的。由于序列对称和反对称的要求，位置向量必关于原点对称，即正负两边一样长。

例 2.3.2：设 $x(n)=u(n+2)-u(n-7)$，将它分解为偶和奇分量。

解：序列 $x(n)$称为矩形序列，它只在$-2<=n<=7$ 区间不为零。下面的程序 hc232 可以求其偶部和奇部。

```
n = [-2:10]; x = stepseq(-2,-2,10)-stepseq(7,-2,10);
[x1,n1] = seqfold(x,n);       % 求出序列 x(-n)
[xe,nxe] = seqadd(0.5*x,n,0.5*x1,n1);
[xo,nxo] = seqadd(0.5*x,n,-0.5*x1,n1);
```

以下的绘图语句略去。

图 2.3.2 清楚地显示了分解的结果。

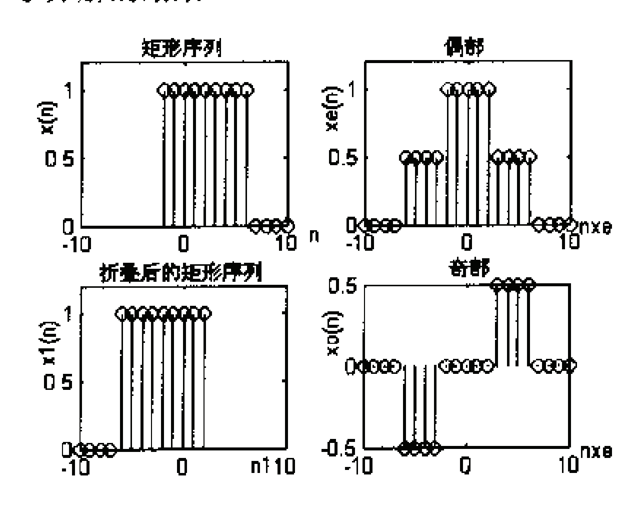


图 2.3.2　实序分解为奇偶序列

2.3.3　复数域的偶序列和奇序列

可以把上述偶序列和奇序列的概念推广到复数域：一个复数序列 $x_e(n)$，如果满足 $x_e(n)=$

$x_e*(-n)$，称为共轭对称序列，或偶序列；如果满足：$x_o(n)=-x_o^*(-n)$，就称为共轭反对称序列，或奇序列。这里 x_e^*，x_o^*分别表示 x_e，x_o 的共轭序列。

定理：任何任意复数序列 $x(n)$可用分解成它的偶和奇分量，见式（2.3.4）。

$$x(n)=x_e(n)+x_o(n) \tag{2.3.4}$$

其中偶部和奇部分别由（2.3.5a）和（2.3.5b）式给出。

$$x_e(n)=\frac{1}{2}\left[x(n)+x^*(-n)\right] \tag{2.3.5 a}$$

$$x_o(n)=\frac{1}{2}\left[x(n)-x^*(-n)\right] \tag{2.3.5 b}$$

它的证明也留给读者自行完成。

只要把对实数序列进行分解的程序中的第一句右端的 x 改成它的共轭 conj(x)；这个程序就同时适用于复数序列的分解。由此可以写成一个求任意序列偶和奇分量的通用子程序 evenodd.m：

```
function [xe,xo,m]=evenodd(x,nx)
[x1,nx1] = seqfold(conj(x),nx);          % 求x1(n)=x*(-n)
[xe,m]=seqadd(x/2,nx,x1/2,nx1)           % 按(2.3.5)求出序列xe
xo=seqadd(x/2,nx,-x1/2,nx1)              % 按(2.3.5)求出序列xo
```

这里，利用了 nxe=nxo 的条件，统一把位置向量称为 m。由于子程序 evenodd 中包含了 seqfold，seqadd 等 MATLAB 的信号处理工具箱没有的的子程序，所以必须把本书的子程序集放在 MATLAB 的搜索路径下，才能运行。

例 2.3.3：求下列序列的结果及图形。

(a)n1=2:8,x1=sin(n1);n2=-1:3;x2= x2=ones(1,5);求 x1 与 x2 之和。

(b) 求 x1 及 x2 之积。

(c)n3=0:4;x3=[6,5,4,3,2]; 求它对 n=0 点折叠后的序列 y3。

(d)利用(b)的结果，构成一个对称序列 y4，使它的右半部等于序列 x3。

其 MATLAB 程序的核心语句如下：

```
n1=2:8,x1=sin(n1);n2=-1:3;x2= ones(1,5);
[y1,ny1]=seqadd(x1,n1,x2,n2);                          % 解a.
[y2,ny2]=seqmult(x1,n1,x2,n2);                         % 解b.
n3=0:4;x3=[6,5,4,3,2]; [y3,ny3]=seqfold(x3,n3);        % 解c.
ny4=-4:4; y4=[y3(1:4),x3]; hold on %                   % 解d.
```

由于 $x3$ 和 $y3$ 在 n=0 的点上都有相同的值，根据按 $x3$ 构成对称的要求，应该删去一个数据，故 $y3$ 的五点中只取了前四点 $y3(1:4)$。以后会指出，对频率特性处理时常要采取删去一个对称中心数据的做法。在四个子图中分别用 stem(ny1,y1),…,stem(ny4,y4)绘图，可以得到图 2.3.3。y4 关于 n=0 对称，所以是偶序列。

如果想构成奇序列，语句应为 `ny5=-4:4; y5=[-y3(1:4),0,x3(2:5)] ; stem(ny5,y5,'*:')`，其图形见同图中的虚

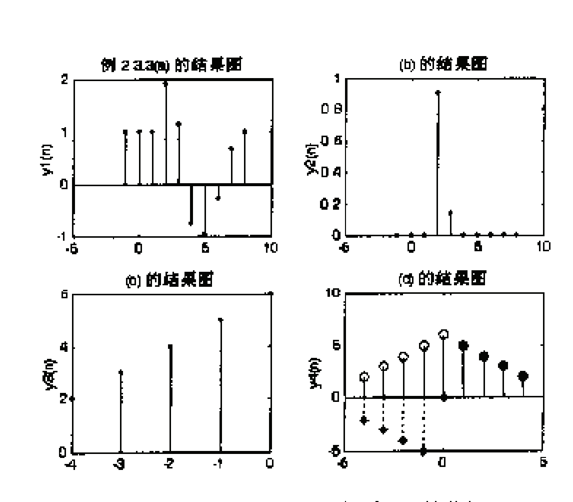


图 2.3.3　例 2.3.3 所得序列的图形

线及*号。注意在 n=0 点处，y5 必须为零，才符合反对称的定义。

2.3.4 序列的卷积

两序列 x 和 h 的卷积 y 定义为：

$$y(n)=\sum_{k=-\infty}^{\infty}x(k)h(n-k)=x(n)\otimes h(n) \tag{2.3.6}$$

卷积运算满足交换律，即：

$$y(n)=\sum_{k=-\infty}^{\infty}h(k)x(n-k)=h(n)\otimes x(n)=x(n)\otimes h(n) \tag{2.3.7}$$

卷积可以用图解的方法来求得，这时 $h(n-k)$要解释为对 k 折叠并移位了 n 的 $h(k)$。输出 $y(n)$是根据 $x(k)$和 $h(n-k)$覆盖区内样本乘积的和求得的。下面用一个例子来说明。

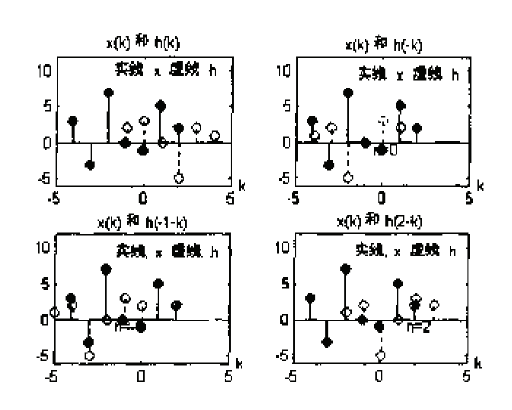


图 2.3.4　卷积序列的折叠和移位

例 2.3.4：给出以下两个序列，试求其卷积 $y(n)=x(n)\otimes h(n)$。

$x=[3,-3,7,0,-1,5,2]$; $nx=[-4:2]$

$h=[2,3,0,-5,2,1]$; $nh=[-1:4]$

解：在图 2.3.4 中给出了四个子图，上左图表示出原始序列 $x(k)$和 $h(k)$，上右图表示出 $x(k)$和把 $h(k)$折叠后的 $h(-k)$。下左图给出 $x(k)$和把 $h(k)$折叠并移位-1 后的 $h(-1-k)$。求和区域$\langle k\rangle$取所有 $x(k)$和 $h(n-k)$有重叠的 k 值范围。于是

$$\sum_{\langle k\rangle}x(k)h(-1-k)=3\times(2)+(-3)\times(-5)+7\times0+0\times3+2\times(-1)=19=y(-1)$$

下右图给出 $x(k)$ 和把 $h(k)$折叠并移位 2 后的 $h(2-k)$。因而得出

$$\sum_{\langle k\rangle}x(k)h(2-k)=7\times1+0\times2+(-1)\times(-5)+5\times0+2\times3=18=y(2)$$

这样得出了 $y(n)$的两个值，$y(n)$的其他值也可用图解求和法类似地求得。注意，$y(n)$的起始点(第一个非零样本点)为 n=-3+(-1)=-4，而终点(末一个非零点)为 n=3+4=7。完整的输出为

$$y=[6,3,5,6,19,-31,30,18,-27,-1,9,2],\quad ny=[-4:7]$$

请读者校验上述结果。注意，产生的序列 $y(n)$比 $x(n)$和 $h(n)$两个序列都要长，$Ny=Nx+Nh-1$。本例中，x 的长度为 7，h 的长度为 6，而 y 的长度为 12=7+6-1，卷积运算可以用前述的序列的基本运算来实现。用 MATLAB 语句表述如下：

```
[h1,nh1]= seqfold(h,nh);                 % 求h1(nh1)=h(-nh)
[h2,nh2]= seqshift(h1,nh1,m);            % 求h2(nh2)=h1(-nh1-m)
[y1,ny1]= seqmult(x,nx,h2,nh2);          % 求y1=h2.*x
y=sum(y1)                                % 求y(m)=sum(h2.*x)
```

用这样的几条语句可以完成对一个位移量 m 的卷积，但它只能帮助概念的理解，用起来很不方便。MATLAB 提供了一个内部函数 conv 来计算两个有限长度序列的卷积。conv 函数假定两个序列都从 n=0 开始。调用方法为：$y=\text{conv}(x,h)$。

函数 conv 不需要输入序列的位置信息。当然，也无法给出输出序列的位置信息。它实际

上是默认 x 和 h 的位置向量 nx 和 nh 都从零开始。如果要对位置向量不是从零开始的两个序列求卷积，需要将函数 conv 稍加扩展。设 x 和 h 为两个位置向量已知的有限序列：{x(n); nx = nxs:nxf}及{h(n); nh=nhs:nhf}。现在要求它们的卷积 $y(n)$及其位置向量 ny。

显然，$y(n)$也是有限序列。从上面的讨论，可知其长度为：

length(y)=length(x)+length(h)-1

而起点和终点分别为：nys =nxs+nhs 及 nyf =nxf+nhf。再利用 MATLAB 中原有的 conv 函数，编出带有 ny 输出的 convwthn 程序如下：

```
function [y,ny] = convwthn (x,nx,h,nh)
nys = nx(1)+nh(1); nyf = nx(end) + nh(end);  % end表示最后一个下标
y = conv(x,h); ny = [nys:nyf];
```

例 2.3.5：用 convwthn 求例 2.3.4 中的卷积。

解：MATLAB 程序 hc235

```
x = [3,-3,7,0,-1,5,2]; nx = [-4:2];
h = [2,3,0,-5,2,1]; nh = [-1:4];
[y,ny] = convwthn (x,nx,h,nh)
```

运行结果为：

```
y =   6   3   5   6  19  -31  30  18  -27  -1  9  2
ny = -5  -4  -3  -2  -1    0   1   2    3   4  5  6
```

在 MATLAB 中还有一种方法可用来求卷积，它利用了矩阵与向量相乘的算法，将在第 2.5.3 节中介绍。

2.3.5　序列的相关

数字信号处理应用中广泛用到相关运算，用它来度量两个序列相似的程度。给出两个长度相同，幅度有限的序列 $x(n)$和 $y(n)$，则 $x(n)$和 $y(n)$的互相关是一个序列 $r_{xy}(k)$，其定义为：

$$r_{xy}(k)=\sum_{n=-\infty}^{\infty}x(n)y(n-k) \tag{2.3.8}$$

下标 k 称为移位或迟延参数，当 $x(n)=y(n)$时，称为自相关。其定义为：

$$r_{xx}(k)=\sum_{n=-\infty}^{\infty}x(n)x(n-k) \tag{2.3.9}$$

它提供了同样序列而对准不同的位置情况下自相似程度的度量。MATLAB 的信号处理工具箱中也提供了一个称为 xcorr 的函数来进行序列的相关计算。它的最简单的形式为 `rxy=xcorr(x,y)`，用于计算向量 x 和 y 之间的互相关，而 rxx=xcorr(x)则用于计算向量 x 的自相关。

同 conv 函数一样，xcorr 函数不能提供时间(或位置)信息，它默认的输入位置向量是：

```
nx=0:length(x)-1; ny=0:length(y)-1;
```

与卷积运算相仿，相关运算的输出序列长度也为两输入序列长度之和减 1，故输出位置向量为：

nrxy=0:length(x)+length(y)-1;

如果输入位置向量不是从零开始，则输出位置向量必须做相应的改变，与求卷积的情况类似。

把卷积运算和相关运算的定义(2.3.6)式和(2.3.8)式做一比较,可以看到两者非常相似。只要把其中一个序列做一次折叠，互相关可以写成如下的卷积形式:

$$r_{xy}(m)= y(m)\otimes x(-m)$$

自相关可以写成如下形式：

$$r_{xx}(m)= x(m)\otimes x(-m)$$

由此可见，如果序列具有有限长度的话，相关可以用 conv 函数来求得。因此也可以调用本书中自编的包含位置向量的卷积函数 convwthn。

例 2.3.6：在本例中，将演示互相关序列的应用。令 $x(n)$ = [3, 5, −7, 2, −1, −3, 2]为一原型序列，令 $y(n)$为 $x(n)$加入噪声干扰并移位后的序列：$y(n)=x(n-2)+w(n)$。

其中 $w(n)$为具有零均值和单位方差的高斯序列。计算 $x(n)$和 $y(n)$之间的互相关。

解：由 $y(n)$的结构可以看出它相似于 $x(n-2)$，因而它们之间的相关应该在 n=2 处显示出最强。为了测试出相关最强的位置，需要同时求出相关的位置向量，而 xcorr 函数不具备这个功能。为了借用带位置向量的卷积函数 convwthn 来计算其互相关，需要在程序中把 x 折叠，再求它与 y 的卷积。

```
x = [3, 5, -7, 2, -1, -3, 2]; nx=[-3:3];      % 给定信号x(n)
[y,ny] = seqshift(x,nx,2);                     % 得到迟延信号 x(n-2)
w = randn(1,length(y)); nw = ny;               % 产生 w(n)
[y,ny] = seqadd(y,ny,w,nw);                    % 得到迟延加扰信号 y(n)
[x1,nx1] = seqfold(x,nx);                      % 把x折叠, 得到 x(-n)
[rxy,nrxy] = convwthn(y,ny,x1,nx1);            % 用带n的卷积求互相关
stem(nrxy,rxy)                                 % 标注语句略去
```

从图 2.3.5 可以看到互相关的峰值确实在 k =2 处。这说明 $y(n)$和 $x(n)$移位 2 后相似。改变其他的噪声序列，并不影响测定的基本结果。说明这种求相关的方法具有较好的抗干扰能力。在雷达信号处理中，计算发射波与带有很大干扰的回波信号之间的相关，可以提高对目标进行距离测定的精确度。

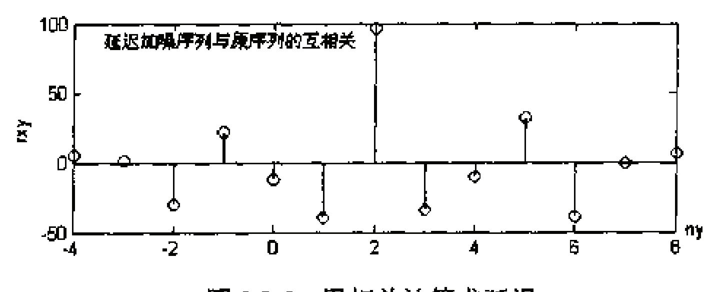


图 2.3.5　用相关计算求延迟

2.3.6　MATLAB 中提供的序列生成函数

在 MATLAB 的信号处理工具箱函数中，还有一些生成特殊序列的函数，见附录 A。在此中举出一些，如表 2.3.1 所示。读者可用 help 命令找到其调用方法，并画出其图形。

表 2.3.1　生成特殊序列函数的举例

函数	说明	函数	说明
chirp	扫频的频率余弦发生器	rectpuls	采样的非周期性的方波发生器
diric	Dirichlet(周期性 sinc)函数	sawtooth	锯齿函数
pulstran	脉冲序列发生器	sinc	Sinc 或者 sin(pi*x)/pi*x 函数
tripuls	非周期性的三角形脉冲发生器	square	方波函数

2.4　时域离散系统

设时域离散（Discrete-Time，DT）系统的输入为 $x(n)$，经过规定的运算，系统输出序列用 $y(n)$表示。若运算关系用 $T[\cdot]$表示，输出与输入之间的关系用（2.4.1）式表示。

$$y(n)=T[x(n)] \tag{2.4.1}$$

在时域离散系统中，最重要最常用的是线性时不变系统，这是因为很多物理过程都可用这类系统表征，且便于分析。

2.4.1　线性系统

满足线性叠加原理的系统称为线性系统（linear system）。设 $x_1(n)$和 $x_2(n)$分别作为系统的输入序列，其输出分别用 $y_1(n)$和 $y_2(n)$表示，即

$$y_1(n)=T\{x1(n)],\quad y_2(n)=T\{x_2(n)]$$

那么线性系统一定满足下面两个公式：

$$T\{x1(n)+x2(n)]=y1(n)+y2(n) \tag{2.4.2}$$

$$T[ax1(n)]=ay1(n) \tag{2.4.3}$$

满足(2.4.2)式称为线性系统的可加性；满足(2.4.3)式称为线性系统的比例性，式中 a 是常数。将以上两个公式结合起来，可表示成

$$y(n)=T[ax_1(n)+bx_2(n)]=ay1(n)+by2(n) \tag{2.4.4}$$

上式中，a 和 b 均是常数。

例 2.4.1：分析下列方程是否为线性系统。

（1）$y(n)=ax(n)+b$；(a 和 b 是常数)

（2）$y(n)=x(n)\sin(\omega_0 n+0.1\pi)$

解：（1）$y1(n)=T[(x1(n))=ax1(n)+b$

$y2(n)=T[x2(n)]=ax2(n)+b$

$y(n)=T[x1(n)+x2(n)]=ax1(n)+ax2(n)+b$

$y(n)\neq y1(n)+y2(n)$

因此，该系统不是线性系统。

（2）用同样方法可以得知第二个式子所代表的系统是线性系统，请读者自己证明。

2.4.2　线性时不变离散系统

如果系统对输入信号的运算关系 $T[\cdot]$在整个运算过程中不随离散时间(此处是自变量 n)变化，或者说系统对于输入信号的响应与信号加于系统的时间无关，则这种系统称为线性时

不变离散系统(Linear Time-Invariant，LTI)，对于离散系统，自变量不是连续时间 t 而是整数位置 n，所以严格地说，应称为线性移不变离散系统(Linear Shift-Invariant，LSI)。不过，出于习惯，还是用“时不变”的名词。用公式表示如下：

$$\left.\begin{aligned} y(n) &= T[x(n)] \\ y(n-n_0) &= T[x(n-n_0)] \end{aligned}\right\} \tag{2.4.5}$$

式中 n_0 为任意整数。检查一个系统是否是时不变系统，就是检查它是否满足(2.4.5)式。

例 2.4.2：检查 $y(n)=ax(n)+b$ 代表的系统是否是时不变系统，上式中 a 和 b 是常数。

解：$y(n)=ax(n)+b$，$y(n-n_0)=ax(n-n_0)+b$，$y(n-n_0)=T[x(n-n_0)]$。

因此该系统是时不变系统。

例 2.4.3：检查 $y(n)=nx(n)$所代表的系统是否是时不变系统。

解：

$$y(n)=nx(n)$$

$$y(n-n_0)=(n-n_0)x(n-n_0)$$

$$T[x(n-n_0)]=nx(n-n_0)$$

可见：$y(n-n_0)\neq T[x(n-n_0)]$

因此该系统不是时不变系统。同样方法可以证明 $y(n)=x(n)\sin(\omega_0 n+0.1\pi)$ 所代表的系统不是时不变系统。

2.4.3 LTI 离散系统的数学描述——线性常系数差分方程

由集总参数（Lumped Parameter）部件构成的 LTI 离散系统（严格地说，这样的系统英文缩写名为 LTIL，但因实际上较少遇到分布参数系统，所以往往仍只写成 LTI 系统）可以用一个线性常系数差分方程表示：

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{i=0}^{M} b_i x(n-i) \qquad a_0 = 1 \tag{2.4.6}$$

式中 $x(n)$ 和 $y(n)$ 分别是系统的输入序列和输出序列。a_k 和 b_i 均为常数向量，对于物理可实现的系统，$N \geqslant M$。如果 $a_N \neq 0$,则方程是 N 阶的。已知系统的输入序列，用这个方程可以根据当前输入 $x(n)$和以前 M 点的输入 $x(n-M),\cdots,x(n-1)$，以及以前 N 点的输出 $y(n-N),\cdots,y(n-1)$来计算当前输出 $y(n)$。大体有以下三种计算方法。

（1）递推算法：把方程(2.4.6)改写为

$$y(n)=\sum_{i=0}^{M} b_i x(n-i)-\sum_{k=1}^{N} a_k y(n-k) \tag{2.4.7}$$

如果开始时刻是n=0,除当前输入$x(0)$外，还需要知道此时刻之前的N个y的初始值 y0= [$y(-N),\cdots,y(-1)$] 及M个x的初始值x0= [$x(-M),\cdots,x(-1)$]，统称为初始条件。此时方程的右端都是已知数，可以求出$y(0)$。得知$y(0)$后，又可进而根据$x(1)$和以前的x和y值求$y(1)$，按时间向前依次递推，就可求出所有要求的输出y。

（2）经典方法：先把方程的解写成通解和特解两部分，如（2.4.8）式。

$$y(n)= y_{\rm H}(n)+ y_{\rm P}(n) \tag{2.4.8}$$

其通解部分 $y_{\rm H}$(n)为

$$y_{\rm H}(n)=\sum_{k=1}^{N} c_k {z_k}^n \tag{2.4.9}$$

其中 $z_k, k=1,2,\cdots,N$ 是特征代数方程

$$\sum_{k=1}^{N} a_k z^k = 0 \tag{2.4.10}$$

的 N 个根。特征方程在确定系统稳定性时是很重要的。如果这些根满足下述条件:

$$|z_k|<1 \qquad k=1,2,\cdots,N \tag{2.4.11}$$

则由(2.4.6)式描述的系统是稳定的。(2.4.9)式中的待定系数 c_k 由初始条件求得。本书将在第 5 章中讨论。特解部分 $y_P(n)$由(2.4.6)式的右端确定，此处从略。

（3）卷积法：当系统的初始条件为零，只考虑输入序列所产生的输出序列时，用卷积法。关于卷积法，后面还有更多的讨论，这里也不展开。

本章着重讨论递推算法，现用例2.4.4来说明。

例 2.4.4：设系统用差分方程 $y(n)-y(n-1)+0.9y(n-2)=x(n)$描述,输入序列为 $x=\delta[n]$,

（1）设初始条件为 $y(-2)=0,y(-1)=0$；（2）设初始条件为 $y(-2)=2,y(-1)=1$。求输出序列 $y(n)$。

解：根据题设：$x(0)=1$，$x(1)=x(2)=\ldots=0$，该系统的差分方程是二阶的，需要两个初始条件。

（1）的递推结果如下：

$y(0)= y(-1)- 0.9y(-2)+x(0)=1$

$y(1)= y(0)- 0.9y(-1)+x(1)=1$

$y(2)= y(1)- 0.9y(0)+x(2)=0.1$

……

（2）的递推结果如下：

$y(0)= y(-1)- 0.9y(-2)+x(0)=1-0.9\times 2+1=0.2$

$y(1)= y(0)- 0.9y(-1)+x(1)=0.2-0.9\times 1+0=0.7$

$y(2)= y(1)- 0.9y(0)+x(2)=-0.7-0.9\times 0.2+0=-0.88$

……

当系统的阶次较高时，手工递推显然很麻烦，且容易出错，需要计算机来协助完成。

2.4.4　用 MATLAB 的递推解法

差分方程递推过程在 MATLAB 中用一个称为 filter 的子程序来实现。调用的最简单形式为

$$y=\text{filter}(b,\ a,\ x)$$

其中：b = [b0, b1,···, bM];　a = [a0, a1, ···, aN]。

这是由 (2.4.6) 式中给出的系数数组，而 x 则是输入序列。求得的输出 y 将和输入 x 的长度一样。系数 a0 必须不为零。对上述例中(1)，假如取 x 的长度为 8，可键入：

```
b=1;a-[1,-1,0.9]; x=[1,zeros(1,7)];
y1=filter(b,a,x)
```

结果为：y=[1.0000 1.0000 0.1000 -0.8000 -0.8900 -0.1700 0.6310 0.7840]。

当初始条件为零，输入为单位脉冲函数时，系统的输出称为此系统的脉冲响应 h（见 2.5 节），所以此处有 $h=y1$。

在有初始条件时，仍可以用filter函数求解。它的调用格式将在第5章中讨论。

2.4.5 用时移算子 z^{-1} 表示的离散系统

在图 2.2.2 中，引用了时移算子的概念，这里再作深一步的探讨。设

$$y(n)=x(n+1) \tag{2.4.12}$$

定义 z 为时移算子，可以写成 $y[n]=z\{x[n]\}$，简写为

$$y(n)=x(n+1)=z\ x(n) \tag{2.4.13}$$

如果 x=[2,4,6,8]，nx=[-1,0,1,2]，则 y=[2,4,6,8]， ny=[-2, -1,0,1]。所以时移算子 z 的作用是把位置向量提前，$n=0$ 时的 y 要取 $n=1$ 时的 x，这意味着要预知未来，理论上可以这样定义，但从物理上是无法实现的。

可以定义 z 的逆运算子 z^{-1}，它的作用是把位置向量推迟。具体说，若 $y1(n)=z^{-1}x(n)$就等价于

$$y1(n)=z^{-1}x(n)=x(n-1) \tag{2.4.14}$$

在上例中意味着 $y1$=[2,4,6,8]，$ny1$=[0,1,2,3]，即 $y(0)=x(-1)$。就是说，把 x 序列延迟一个节拍可以得到 y，这在物理上是可以实现的。这个算子不仅是有抽象的数学意义，同样有着非常实际的可操作性。延迟线可以起这个作用，计算机中的移位寄存器、程序中的按指令节拍进行的转存等都可以等价于迟延功能。因此 z^{-1} 是数字信号处理中的一种基本运算。数字滤波器和计算机中就是用乘、加和迟延三种运算来组成的，后面将会在频域来探讨 z 变换和 z^{-1} 算子在频域的含义，但不妨先从时域来理解它。

z^{-1} 是延迟一拍的运算，因而 z^{-k} 就是把已有信号延迟 k 拍。有

$$x(n-k)=z^{-k}x(n), \tag{2.4.15}$$

于是离散线性系统的差分方程(2.4.6)就可以表示为

$$\sum_{k=0}^{N}a_k z^{-k}y(n)=\sum_{i=0}^{M}b_i z^{-i}x(n) \qquad a_0=1 \tag{2.4.16}$$

要求出 $y(n)$，可以用移项的方法，注意 $a0$=1，得到

$$y(n)=\sum_{i=0}^{M}b_i z^{-i}x(n)-\sum_{k=1}^{N}a_k z^{-k}y(n) \tag{2.4.17}$$

虽然等式右边还有 $y(n)$,但注意它前面都有 z^{-k}，且 k 从 1 开始，也就是最右端的求和号内，都是过去的 $y(n)$，因而是已知数。这个式子就是以 z^{-1} 算子表示的递推公式。

从形式上推理，是不是可以把（2.4.16）式中左端的算子除以右端的呢？那样就成为

$$y(n)=\frac{\sum_{i=0}^{M}b_i z^{-i}}{\sum_{k=0}^{N}a_k z^{-k}}x(n)=H(z)x(n) \qquad a_0=1 \tag{2.4.18}$$

式中：

$$H(z)=\frac{b_0+b_1z^{-1}+\cdots+b_{N-1}z^{-M+1}+b_Nz^{-M}}{a_0+a_1z^{-1}+\cdots+a_{N-1}z^{-N+1}+a_Nz^{-N}} \tag{2.4.19}$$

由于对 z 算子多项式的除法没有定义，因而(2.4.18)式也没有计算的办法。这里只是把它作为一种形式或符号来对待的。它所体现的运算要在第 4 章学了 z 变换以后才能搞清。上述的算子符号 $H(z)$，就代表了系统差分方程。(2.4.18)式表示，用算子 $H(z)$作用于输入序列 $x(n)$

就得出输出序列 $y(n)$，所以 $H(z)$也称为 LTI 离散系统的传递函数或系统函数。在这里只给出一个形式上的表述，更严格的理论将在第 5 章中讨论。

有了这个算子表达式的好处是：不必通过复杂的 z 变换的概念就可以写出系统的结构图和信号流图，并解出系统的传递出数。

2.5　脉冲响应——用时间序列表征的系统特性

2.5.1　脉冲响应的特性

设系统的输入为脉冲序列 $x(n)=\delta(n)$，输出 $y(n)$的初始状态为零，把这种条件下的系统输出定义为系统的（单位）脉冲响应 $h(n)$。用公式表示为

$$h(n)=T[\delta(n)] \tag{2.5.1}$$

$h(n)$和模拟系统中的单位冲激响应 $h(t)$相类似。脉冲响应可以由系统的差分方程惟一地解出，两者有一一对应的关系，所以可用它来代表系统的时域特征。知道了系统的脉冲响应，如同知道了系统的差分方程一样，可以根据系统的输入求出它的输出。计算的方法就是前面已经提到的卷积。

设系统的输入用 $x(n)$表示，将它表示成单位脉冲序列移位加权和：

$$x(n)=\sum_{m=-\infty}^{\infty}x(m)\delta(n-m) \tag{2.5.2}$$

那么系统输出为 $y(n)=T\left[\sum_{m=-\infty}^{\infty}x(m)\delta(n-m)\right]$。根据线性系统的叠加性质 $y(n)=\sum_{m=-\infty}^{\infty}x(m)T\left[\delta(n-m)\right]$，将 $h(n)=T[\delta(n)]$代入，根据时不变性质得到

$$y(n)=\sum_{m=-\infty}^{\infty}x(m)h(n-m)=x(n)\otimes h(n) \tag{2.5.3}$$

(2.5.3)式表示：线性时不变系统的输出等于输入序列和该系统的脉冲响应的卷积。只要知道系统的脉冲响应，根据任意输入的 $x(n)$，都可以求出系统的输出 $y(n)$。

除了可以像上例那样，在系统中输入脉冲序列，并用 filter 函数解出脉冲响应外，在 MATLAB 中还有求离散系统脉冲响应的专门函数 impz。它有多种调用方式。基本形式为：

[h,t]=impz(b,a,L)

其中，b 和 a 为系统差分方程(2.4.6)右端和左端的系数向量，L 为要求的点数，即所要求的脉冲响应的长度。输出变元 h 为这些点上的脉冲响应值，t 是对应的时刻。

键入[h,t]=impz(b,a)，程序将自动选择脉冲响应的长度，直到 h 的值可以忽略。其长度由输出 t 向量的值可以看到。

如果不给左端变量，键入 impz(b,a),程序将自动给出脉冲响应的曲线,但不给数据。

因此求例 2.4.4 中系统的脉冲响应时，可以用下列 MATLAB 程序：

```
b=[1, 0, 0];a=[1,-1,0.9]; [h,t]=impz(b,a,L)
```

若取 L=7,则此程序运行结果如下，与前面的结果相同。

```
h =        1.0000          t =      0
           1.0000                   1
           0.1000                   2
          -0.8000                   3
          -0.8900                   4
          -0.1700                   5
           0.6310                   6
```

2.5.2 用解析卷积方法求系统输出

可以有很多方法来求卷积(2.5.3)式。如果输入序列和脉冲响应序列都是数学函数(有限或无限长度)，则可用分析的方法求卷积，得到对所有的 n 都适用的 $y(n)$函数形式。

例 2.5.1：将例 2.4.4 中的矩形脉冲序列 $x(n)=u(n)-u(n-10)$作为对一个脉冲响应为 $h(n)=(0,9)^n u(n)$的 LTI 系统的输入，求输出 $y(n)$。

解：输入 $x(n)$及脉冲响应 $h(n)$示于图 2.5.1 中。

$$y(n)=\sum_{n=0}^{9}0.9^{(n-k)}u(n-k)=(0.9)^n\sum_{n=0}^{9}0.9^{-k}u(n-k) \tag{2.5.4}$$

(2.5.4)式中的总和大体上是一个几何级数和，其中 $u(n-k)$项将依赖于不同的 n 和 k，可能在以下三种不同的情况下取值:

（1）$n<0$：此时 $u(n-k)=0$，$0\leqslant k\leqslant 9$。在此情况下 $x(n)$和 $h(n)$的非零值不互相覆盖。因而从(2.5.4)式得

$$y(n)=0 \tag{2.5.5}$$

（2）$0\leqslant n<9$：则 $u(n-k)=1$，$0\leqslant k\leqslant n$。在此情况下，脉冲响应 $h(n)$与输入 $x(n)$ 部分覆盖。因而由(2.2.4)式得（2.5.6）式：

$$\begin{aligned}y(n)&=(0.9)^n\sum_{k=0}^{n}0.9^{-k}u(n-k)=(0.9)^n\sum_{k=0}^{n}\left[0.9^{-1}\right]^k\\&=(0.9)^n\frac{1-0.9^{-(n+1)}}{1-0.9^{-1}}=10[1-0.9^{n+1}]\ 0\leqslant n<9\end{aligned} \tag{2.5.6}$$

（3）$n\geqslant 9$：故 $u(n-k)=1, 0\leqslant k\leqslant 9$，在此情况下，$h(n)$完全与 $x(n)$覆盖。代入(2.5.4)式得(2.5.7)式。

$$y(n)=(0.9)^n\sum_{k=0}^{9}0.9^{-k}=(0.9)^n\frac{1-0.9^{-10}}{1-0.9^{-1}}=10(0.9)^{n-9}[1-0.9^{10}]\ \ n>9 \tag{2.5.7}$$

由(2.5.5)式、(2.5.6)式和(2.5.7)式给出了完整的响应式。它表示于在图 2.5.1 中，$y(n)$由三段组成，显示了输入脉冲方波经过系统后的输出发生了畸变。

演示程序 fgp251.m 给出了任意两个解析序列进行卷积的动态过程。

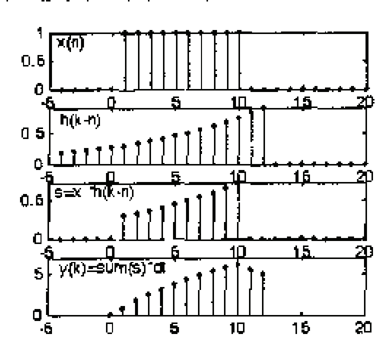


图 2.5.1　卷积的图示

2.5.3　用向量-矩阵乘法进行卷积计算

当输入序列 x 和脉冲响应序列 h 都是以数值给出时，如例 2.3.4 那样，可以用 MATLAB 中的卷积函数 conv 来计算。conv 是一个内部函数，读者无法看到它的实现原理和方法。这里介绍一种向量—矩阵相乘的卷积计算方法。虽然它是一种计算效率较低的算法，但从中可以清楚地看懂其计算的原理，现说明如下：

按定义，卷积计算的公式为 2.5.1 节所述的(2.5.3)式：

$$y(n)=\sum_{m=-\infty}^{\infty}x(m)h(n-m)=x(n)\otimes h(n)\qquad n=1,2,\cdots,L$$

假如 $x(m)$和 $h(m)$分别为具有有限长度 Nx 和 Nh 的行向量，则 $L=Nx+Nh-1$。不失一般性，对于所取的任一 n，可以把卷积式写成一个行向量和一个列向量的乘积：

$$y(n)=\sum_{m=1}^{Nx}x(m)h(n-m)=[x(1),x(2),\cdots,x(Nx)]*\begin{bmatrix}h(n-1)\\h(n-2)\\\vdots\\h(n-Nx)\end{bmatrix}\qquad n=1,2,\ \cdots,L$$

把 n 逐次加一，并把等式左右的 $y(n)$和 h 向量沿列向排列，注意 $n<0$ 时，$h(n)=0$。可得：

$$[\,y(1),y(2),\cdots,y(L)\,]=[x(1),x(2),\cdots,x(Nx)]\otimes\begin{bmatrix}h(1-1)&h(2-1)&\cdots&h(L-1)\\0&h(2-2)&\cdots&h(L-2)\\\vdots&\vdots&\vdots&\vdots\\0&0&\cdots&h(L-Nx)\end{bmatrix}\quad(2.5.8\ \text{a})$$

因而卷积计算可写成如下的 MATLAB 乘积形式

$$y=x*H\qquad(2.5.8\ \text{b})$$

其中：$y=[\,y(1),y(2),\cdots,y(L)\,]$，$x=[x(1),x(2),\cdots,x(Nx)]$。

$$H=\begin{bmatrix}h(1-1)&h(2-1)&\cdots&h(L-1)\\0&h(2-2)&\cdots&h(L-2)\\\vdots&\vdots&\vdots&\vdots\\0&0&\cdots&h(L-Nx)\end{bmatrix}\qquad(2.5.9)$$

下面用例 2.3.4 作为蓝本，观察(2.5.9)式中 H 矩阵的组成方法。从(2.5.9)式很易得知，H 矩阵有 Nx 行和 $L=Nx+Nh-1$ 列。题中的数据为

$x=[3,-3,7,0,-1,5,2]$; h = [2,3,0,-5,2,1];

它们的长度为 $Nx=7$，$Nh=6$，故 $L=6+7-1=12$，所以矩阵 H 为

$$H=\begin{bmatrix}2&3&0&-5&1&1&0&0&0&0&0&0\\0&2&3&0&-5&1&1&0&0&0&0&0\\0&0&2&3&0&-5&1&1&0&0&0&0\\0&0&0&2&3&0&-5&1&1&0&0&0\\0&0&0&0&2&3&0&-5&1&1&0&0\\0&0&0&0&0&2&3&0&-5&1&1&0\\0&0&0&0&0&0&2&3&0&-5&1&1\end{bmatrix}\qquad(2.5.11)$$

算出：$x*H=$ 　6　3　5　6　19　-31　30　18　-27　-1　9　2

这和前面得出的卷积结果相同，说明这样取 $x1$ 和 H 是正确的，现在看看它的规律。

在 H 的第一行上，前六项是脉冲过渡函数的折叠，即 fliplr(h)，后面补以六个零，即补零的数目等于 x 的长度 Nx 减一。向下各行则依次右移一位，直到整个 fliplr(h)移到右端。这样 H 矩阵的列数为 $L=Nx+Nh-1$。而它的行数为 Nx，在本例中 H 是 7×12 阶矩阵。H 矩阵的列数实际上就是卷积输出序列 y 的长度。这种依次下移并右移一位的矩阵称为 Toeplitz 矩阵。只要知道它的第一列向量 C^T 和第一行向量 R，就可以调用 MATLAB 函数 H=toeplitz(C，R)来生成。

例如取 C=[1:5]; R=[5:10];键入 H=toeplitz([1:5],[5:10])，得到

$$H=\begin{bmatrix} 1 & 6 & 7 & 8 & 9 & 10 \\ 2 & 1 & 6 & 7 & 8 & 9 \\ 3 & 2 & 1 & 6 & 7 & 8 \\ 4 & 3 & 2 & 1 & 6 & 7 \\ 5 & 4 & 3 & 2 & 1 & 6 \end{bmatrix}$$

注意对角线上的项服从 C 优先的原则，所以 R 中的第一个元素在矩阵中没有反映。

根据上述规则，在本例中 H 矩阵可按其第一列的转置向量 C 和第一行 R 向量生成：

```
C=[h(1),zeros(1,Nx-1)];R=[h,zeros(1,Nx-1);
H=toeplitz(C,R)
```

所以用向量-Toeplitz 矩阵相乘方法求卷积的程序 hc252 核心语句为：

```
x=input('x=');h=input('h=');              % 输入x 和h 向量
Nx=length(x);Nh=length(h);L=Nx+Nh-1;     % 求向量长度
% 生成toeplitz 矩阵H
H-toeplitz([h(1),zeros(1,Nx-1)],[h,zeros(1,Nx-1)])
y=x*H      % 用向量-toeplitz 矩阵相乘求卷积
```

运行此程序，并按程序的提示依次输入 x 和 h 序列，就可以得到 toeplitz 矩阵 H 和卷积的结果。

这方法虽然不如直接调用 y=conv(x,h)那样方便，而且运算速度也慢。但 conv 函数是 MATLAB 的内部函数，读者无法从中看到其编程的思路。向量一矩阵相乘方法可以给出非常清楚的概念，并且对于自己编写与卷积有关的程序很有用。例如第 3 章将要讲到的循环卷积，MATLAB 中没有相应的现成公式，conv 函数又用不上，这时向量-矩阵相乘可能是最易掌握的方法了。

如果 y 和 x 均为列向量，则将(2.5.8)式两端取转置，得到

$$y=(H')*x=H_1*x \tag{2.5.10}$$

其中：

$$H_1=\begin{bmatrix} h(0) & 0 & \cdots & & 0 \\ h(1) & h(0) & \cdots & & 0 \\ h(2) & h(1) & h(0) & \cdots & \vdots \\ \vdots & \vdots & \vdots & & \\ h(L-1) & h(L-2) & \cdots & & H(L-Nx) \end{bmatrix} \tag{2.5.11}$$

H_1 的生成方法留给读者思考。

2.5.4 用脉冲响应判断系统的因果性

如果系统在 n 时刻的输出，只取决于 n 时刻及 n 时刻以前的输入序列，而与 n 时刻以后的输入序列无关，则称该系统具有因果性质，或称该系统为因果系统。如果 n 时刻的输出还取决于 n 时刻以后的输入序列，这就违背了因果性原则。此系统是无法实现的，被称为非因果系统。因此系统的因果性是系统的可实现性的一个主要要求。因果系统的脉冲响应必须满足

$$h(n)=0 \qquad n<0 \tag{2.5.12}$$

这从概念上也容易理解，在 $n=0$ 时刻以前，没有加入信号，输出只能等于零。满足(2.5.12)式的序列称为因果序列。在信号处理中，除非有专门的说明，总是假定系统是因果的。

2.5.5 用脉冲响应判断系统的稳定性

所谓稳定的系统，是指系统在有界的输入激励下，其输出也是有界的（Bounded-Input Bounded-Output，BIBO）。系统稳定的充分必要条件是系统的脉冲响应绝对可加，那就是：对于一个因果系统，总可以找到一个足够大的正数 B_h，使得

$$\sum_{n=0}^{\infty}|h(n)| \leqslant B_h < \infty \tag{2.5.13}$$

证：先证明充分性。即满足(2.5.13) 式的系统必然稳定。因为

$$y(n)=\sum_{k=0}^{\infty}h(k)x(n-k)$$

有

$$|y(n)| \leqslant \sum_{k=0}^{\infty}h(k)x(n-k)$$

因为输入序列 $x(n)$有界，即 $|x(n)|<B_x$， $-\infty<n<\infty$， B_x 为任意正数，因此

$$|y(n)| \leqslant B_x\sum_{k=0}^{\infty}|h(k)| \leqslant B_x \cdot B_h \tag{2.5.14}$$

可见输出 $y(n)$是有界的。

再证明其必要性。如果 $h(n)$不满足(2.5.13)式，即对于给出的任意一个大的正数 M，总存在着一个 n_1 使得

$$\sum_{n=0}^{n_1}|h(n)| \geqslant M \tag{2.5.15}$$

那么总可以找到一个或若干个有界的输入，它会引起无界的输出，例如取 $x(n_1-k)=\text{sign}[h(k)]$。

其中 sign 为符号函数，它的取值为±1，即总是按括号内函数 $h(k)$的正负号取值。显然，输入 x 是有界的。取 $n=n_1$ 处，求其引起的输出，根据(2.5.15)式，输出为

$$y(n_1)=\sum_{k=0}^{n_1}h(k)x(n_1-k)=\sum_{k=0}^{n_1}|h(k)| \geqslant M$$

上式说明 $n=n_1$ 时刻的输出 y 为无界，系统不稳定。这也就证明了(2.5.13)式条件的必要性。

例 2.5.2：设线性时不变系统的脉冲响应为

$$h(n)=\begin{cases}a^n u(n) & n\geqslant 0\\ 0 & n<0\end{cases} \tag{2.5.16}$$

式中 a 是实常数，试分析该系统的因果性和稳定性。

解：由于 $n<0$ 时，$h(n)=0$，所以系统是因果系统。下面分析其脉冲响应是否绝对可加。

$$\sum_{n=-\infty}^{\infty}|h(n)|=\sum_{n=0}^{\infty}|a|^n=\lim_{N\to\infty}\sum_{n=0}^{N-1}|a|^n=\lim_{N\to\infty}\frac{1-|a|^N}{1-|a|} \tag{2.5.17}$$

只有当 $|a|<1$ 时

$$\sum_{n=-\infty}^{\infty}|h(n)|=\frac{1}{1-|a|} \tag{2.5.17}$$

有界。因此系统稳定的条件是 $|a|<1$；不满足这个条件，系统就不稳定。系统稳定时，$h(n)$ 的模值随 n 加大而减小，此时序列 $h(n)$ 称为收敛序列。如果系统不稳定，$h(n)$ 的模值随 n 加大而增大，则 $h(n)$ 为发散序列。

例 2.5.3：设系统的脉冲响应 $h(n)=\mu(n)$，当 $n\geqslant 0$。任意输入序列 $x(n)$ 从 $n=0$ 时开始作用于系统。求输出 $y(n)$，并检验系统的因果性和稳定性。

解：$h(n)=\mu(n)$，$y(n)=x(n)^{*}h(n)=\sum_{k=-\infty}^{\infty}x(k)\mu(n-k)$

由于 $k<0$ 时，$x(k)=0$。又当 $n-k<0$ 时，$\mu(n-k)=0$；$n-k\geqslant 0$ 时，$\mu(n-k)=1$。因此，求和的范围为 $0\leqslant k\leqslant n$，即

$$y(n)=\sum_{k=0}^{n}x(k) \tag{2.5.18}$$

上式表示该系统是一个累加器，它将输入序列从加上之时开始，逐项累加，一直加到 n 时刻为止。这相当于连续系统中的积分器。再分析该系统的稳定性，对于任意大的数 M，总可以找到一个 n_1，使得 $n_1>M/1$，此时有

$$\sum_{n=-\infty}^{n_1}|h(n)|=\sum_{n=0}^{n_1}|u(n)|=n_1\cdot 1\geqslant M \tag{2.5.19}$$

因此该系统是一个不稳定系统。

2.5.6 用脉冲响应分类数字滤波器

滤波器是用来对信号进行处理的系统的通用名称。因此离散时间 LTI 系统也可统称为数字滤波器。它可分为以下两类：

（1）有限长度脉冲响应(Finite Impulse Response -FIR)滤波器

如果一个 LTI 系统的单位脉冲响应长度有限,则此系统称为有限长度脉冲响应(或 FIR)滤波器。因此对一个 FIR 滤波器,在 $n<n_1$ 和 $n>n_2$ 时，$h(n)=0$。差分方程(2.4.6)中的下列形式描述了一个因果的 FIR 滤波器。

$$a_0 y(n)=\sum_{i=0}^{M}b_i x(n-i) \tag{2.5.20}$$

不难看出，若 $a_0=1$，该滤波器的脉冲响应就等于等式右端的系数向量。$h(0)=b0, h(1)=b1,\cdots,h(M)=b_M$，而所有其他的 $h(n)$ 为 0。这相当于把当前时刻 n 和以前的 M 点的 x 值加权平均而求得 y，这些加权值就是滤波器的系数。随着 n 的增加，这 $M+1$ 个点是不断向前移

动的，因此，FIR 滤波器也称为滑动平均(MA-Moving Average)滤波器。FIR 滤波器既可用脉冲响应的值 $h(n)$，也可用差分方程右端的系数 b 表示。在 $a_0 \neq 1$ 时，$h=b/a_0$。

要实现 FIR 滤波器，既可用函数 conv(x,h)，也可用函数 filter(b,1,x)。需要注意这两种方法的输出有一个差别：由 y=conv(x,h)函数产生的输出 y 的长度为 x 和 h 序列长度之和减一，即 $Ny=Nx+Nh-1$。因此与输入时刻对应的 y 究竟是哪一点还需要仔细探讨；而由 filter(b,1,x)函数产生的输出序列 y 的长度恰好与 $x(n)$序列的长度相同，每一个输出元素对应着同一位置的输入元素，不会搞错。所以在计算输出信号的问题上，应当尽量使用 filter 函数。

（2）无限长度脉冲响应(Infinite Impulse Response -IIR)滤波器

如果一个 LTI 系统的脉冲响应具有无限长度，则称此系统为无限长脉冲响应(或 IIR)滤波器。这时差分方程(2.4.6)的左端有多项，而右端只有 b_0 项，即

$$\sum_{k=0}^{N} a_k y(n-k) = b_0 x(n) \qquad a_0 \neq 0 \tag{2.5.21}$$

可以把(2.5.21)写成

$$y(n) = b_0 x(n) - \sum_{k=1}^{N} a_k y(n-k) \tag{2.5.22}$$

它描述了一个递归滤波器，其中输出 $y(n)$可用其以前算得的 y 值递推计算而得，所以也称为自回归(AR-Auto Regressive)滤波器。如果考虑脉冲响应，则当 $n>0$ 时，$x(n)=0$，因而输出 $y(n)$是其前面的 N 个点 $y(n-k)(k=1,2,\cdots,N)$的线性组合。只要前面的点有值，后面的点就不可能都为零。可见这样的滤波器具有无限长的脉冲响应，称为 IIR 滤波器。当通用差分方程(2.4.6)左右的系数向量都不只一项时：

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{i=0}^{M} b_i x(n-i) \qquad a_0 = 1$$

它所描述的仍是一个 IIR 滤波器，因为其脉冲响应仍具有无限长度。它由两个部分组成：由系数向量 a 描述的 AR 部分和由系数向量 b 描述的 MA 部分。这样的滤波器也称为自回归滑动平均滤波器，或 ARMA 滤波器。可以用 y=filter(b,a,x)计算滤波器输出。

2.6　序列运算的 MATLAB 函数列表

为了便于查阅，将序列运算所用的 MATLAB 函数归纳为表 2.6.1。

表 2.6.1　一些常用的序列运算及其 MATLAB 表述

运　算	数学形式	MATLAB 表述或子程序
两序列相加	$y(n)=x_1(n)+x_2(n)$	[y,ny]=seqadd(x1,nx1,x2,nx2)
两序列相乘(加窗)	$y(n)=x_1(n)x_2(n)$	[y,ny]=seqadd(x1,nx1,x2,nx2)
序列累加(与积分类似)	$y(n)=\sum_{i=ns}^{n} x(i)$	y=cumsum(x)
右移位 m	$y(n)=x(n-m)$	[y,ny]=seqshift(x,nx,m) 或 y=x; ny=nx-m
对 n=0 点折叠	$y(n)=x(-n)$	[y,ny]=seqfold(x,nx) y = fliplr(x); ny =fliplr(-nx)

（续表）

运　算	数学形式	MATLAB 表述或子程序
长 M 的周期延拓	$y(n)=x((n)_M)$	ny=nsy:nfy; y=x(mod(ny,M)+1);
两序列的卷积	$y(n)=x_1(n)\otimes h_1(n)$	y=conv(x1,x2),或 [y,ny]=convwthn(x1,nx1,x2,nx2)
序列的能量	$E=\sum_{n=ns}^{nf}x(n)x^*(n)$	E=x*conj(x)' 或 E=sum(abs(x).^2)
两序列的相关	$y(m)=\sum_{n=ns}^{nf}x_1(n)x_2(n-m)$	y=xcorr(x1,x2) 或 y=conv(x1,fliplr(x2))
序列的傅里叶变换	—	X=fft(x,N)
序列通过线性系统	差分方程求解	y = filter(b,a,x)

复习思考题

2.1 什么是连续信号的采样序列？其数学定义是什么？在什么条件下可以用它来分析原连续信号的特性？为什么说它与原连续信号在物理上不等价？

2.2 什么是连续信号的等价序列？其数学定义是什么？为什么说它与原连续信号在物理上近似等价？它与采样序列的区别是什么？为什么数字信号处理中用采样序列而不用等价序列作为分析的模型？

2.3 以连续量 t 为自变量的采样序列如何表达？用一个实际的采样开关和 A/D 变换器采集信号的过程来解释这个表达式。

2.4 以离散量 n 为自变量的采样序列如何表达？为什么必须用两个向量 x 和 nx 才能完整地表述一个时间序列。

2.5 在如图 2.2.2 所示的六种序列运算中，可以物理实现的线性运算是哪几种？

2.6 两序列进行相加时，位置向量起什么作用？在子程序 seqadd 中如何体现？

2.7 何谓周期序列，如何由单个周期的序列延拓为多周期序列？如何在多周期序列中取出其单个周期？

2.8 何谓对称序列和反对称序列？为什么说任何（包括实数和复数）序列必定可以分解为偶序列和奇序列之和？如何分解？

2.9 两序列之间卷积的定义是什么？它有什么用处？如何把卷积运算表示为向量和矩阵的乘积？这种表示方法有什么好处？如何使卷积的输出中包括位置向量？

2.10 两序列之间相关的定义是什么？它有什么用处？怎么能把相关运算表示为卷积运算？这有什么好处？

2.11 什么样的系统才能用差分方程来描述？差分方程左右端的系数向量各有什么意义？如何根据系数向量来判别系统属 FIR 或 IIR 类型？

2.12 什么是脉冲响应？什么是差分方程？什么是传递函数？它们的关系式是什么？描述非线性或时变系统时，可以应用此关系式吗？

2.13 如何定义因果性？为什么因果性是每个物理系统的固有性质？脉冲响应是因果的条件是什么？

2.14 有界输入有界输出（BIBO）稳定性的定义是什么？怎样检查系统的稳定性？

2.15 计算脉冲响应有哪些方法？在 MATLAB 中有哪些函数可用？

2.16 移位算子 z 和 z^{-1} 作用于序列 $x(n)$ 时，输出有什么不同？哪个算子是物理可实现的？

习题

2.1 已知位置向量同为 n=-3:3 的以下三个序列：

x=[2,-1,4,7,0,-3,5]；y=[-2,8,0,-3,6,2,-4]；w=[7,0,-9,-3,4,2,-1]；

请计算：(a) $u=x+y$　　(b) $v=x\cdot w$　　(c) $z=x-y\cdot w$

2.2 把题 2.1 中 x, y, w 分解成偶序列和奇序列。

2.3 令 $x(n)=[1,-2,4,6,-5,8,10]$，产生并画出下列序列的样本。

$x_1(n)=3x(n+2)+x(n-4)-2x(n)$

2.4 把题 2.3 中所给的序列分解为偶和奇分量，用 stem 函数画出这些分量。

2.5 求出下列序列的自相关序列 $r_{yx}(k)$和互相关序列 $r_{xy}(k)$，观察出它的特点。

$x(n)=(0.9)^n, 0\leqslant n\leqslant 20; y(n)=(0.8)^{-n}; -20\leqslant n\leqslant 0$

2.6 一个特定的线性和时不变系统,描述它的差分方程如下：

$y(n)+0.1y(n-1)-0.06y(n-2)=x(n)-2x(n-1)$

求系统脉冲响应的前 10 个样本。

(a) 确定系统的稳定性；

(b) 在 $0\leqslant n\leqslant 10$ 之间求得并画出系统的单位脉冲响应，从脉冲响应确定系统的稳定性；

(c) 如果此系统的输入为 $x(n)=[5+3\cos(0.2\pi n)+4\sin(0.6\pi n)]\ \mu(n)$，　在 $0\leqslant n\leqslant 20$ 间求出 $y(n)$的响应。

2.7 设 $g(n)$为偶序列，$h(n)$为奇序列，问以下序列是偶序列还是奇序列？

(a) $x_2(n)=g(n)\bullet g(n)$　　(b) $x_2(n)=g(n)\bullet h(n)$　　(c) $x_3(n)=h(n)\bullet h(n)$

2.8 以下诸序列中，哪些是有界序列？

(a) $x_1(n)=A\alpha^n$,A和 α均为复数，且$|\alpha|<1$

(b) $x_2(n)=A\alpha^n\mu(n)$,A和 α均为复数，且$|\alpha|<1$[$\mu(n)$为单位阶跃序列]

(c) $x_3(n)=3\cos^3(\omega_0 n^2)$

2.9 (a) 设已知一个因果的实序列 $x(n)$在 $n\geqslant 0$ 区域的偶序列部分 $x_e(n)$,试求出原实序列 $x(n)$。如果已知其奇序列部分 $x_o(n)$，也能得知原序列吗？

(b) 如果原序列是一个因果的复序列 $x(n)$，能不能同样做到？

2.10 在某音乐厅中,原始音频信号会由于墙壁和天花板的反射而产生回音。听众所感受的音频信号是 $x(n)$和它回音的合成，令：

$$y(n)=x(n)+\alpha x(n-k)$$

其中：k 是总迟延的脉冲个数；α 是其相对强度。现用相关分析来估计该迟延。

(a) 用解析方法由自相关 $r_{yy}(m)$确定自相关 $r_{xx}(m)$。

(b) 令 $x(n)=\cos(0.2\pi n)+0.5\cos(0.6\pi n)$, $\alpha=0.1$ 及 k=50，产生 200 个 $y(n)$样本，并求出其自相关。从观察 $r_{yy}(m)$中能否得到 α 和 k?

2.11 (a)解析地证明线性卷积具有以下特性:

$x_1(n)\otimes x_2(n)=x_2(n)\otimes x_1(n)$　　:交换律

$[x_1(n)\otimes x_2(n)]\otimes x_3(n)=[x_1(n)\otimes x_2(n)]\otimes x_3(n)$　　:结合律

$x_1(n)\otimes[x_2(n)+x_3(n)]=x_1(n)\otimes x_2(n)+x_1(n)\otimes x_3(n)$:分配律

$x(n)\otimes\delta(n-n_0)=x(n-n_0)$:同一律

(b) 对下面三个序列，用 convwthn 函数,验证上述特性。

$x_1(n)=n[\mu(n+10)-\mu(n-20)]$

$x_2(n)=\cos(0.1\pi n)[\mu(n)-\mu(n-30)]$

$x_3(n)=(1.2)^n[\mu(n+5)-\mu(n-10)]$

2.12 令 $x(n)=(0.8)^n\mu(n)$

(a) 解析地求 $x(n)\otimes x(n)$；

(b) 用 conv 函数求出 $x(n)\otimes x(n)$ 的前 20 个样本，将结果与(a)部分的结果相比较。

2.13 设 x=[5,3,-2,0,-3]，h=[4,-3,2,-1]，列出用向量矩阵乘积求卷积的方程 $y=x\otimes h=x*H$ 中的矩阵 H 的表达式，并列出产生此矩阵的 MATLAB 语句，计算其结果。

2.14 设 LTI 系统的脉冲响应为 h=[4,3,2,-1]；卷积输出为 y=[20,-5 ,7 ,-18, 2,-9,3];求输入序列 $x(n)$。

2.15 证明如下的关系式：$\delta(n)=(1-z^{-1})\mu(n)$，并说明$(1-z^{-1})$为何相当于一阶差分运算。

2.16 序列 $x(n)$在 n 处的二阶导数通常可由 $y(n)=x(n+1)-2x(n)+x(n-1)$差分运算式近似。如果 $x(n)$和 $y(n)$表示一个离散系统的输入和输出，问此系统是线性的吗，是时不变的吗？是因果的吗？

2.17 设 LTI 系统的差分方程为下式，求得并画出系统的脉冲响应。

$$y(n)+0.1y(n-1)-0.06y(n-2)=x(n)-2x(n-1)$$

2.18 一个特定的线性和时不变系统,描述它的差分方程如下:

$$y(n)-0.5y(n-1)+0.25y(n-2)=x(n)+2x(n-1)+x(n-3)$$

(a) 确定系统的稳定性；

(b) 在 $0\leqslant n\leqslant 10$ 之间求得并画出系统的脉冲响应.从脉冲响应确定系统的稳定性.

(c) 如果此系统的输入为 $x(n)=[5+3\cos(0.2\pi n)+4\sin(0.6\pi n)]\,\mu(n)$，在 $0\leqslant n\leqslant 20$ 间求出 $y(n)$的响应.

2.19 一个简单的数字微分器表为公式，$y(n)=x(n)-x(n-1)$

请计算输入信号的后向一阶差分。对下列序列执行这个微分并画出其结果，评论实现这个简单的微分器的可用性。

(a) $x(n)=5[u(n)-u(n-20)]$:矩形脉冲

(b) $x(n)=n[u(n)-u(n-10)]+(20-n)[u(n-10)-\mu(n-20)]$:三角脉冲

(c) $x(n)=\sin(\pi n/25)[u(n)-u(n-100)]$:正弦脉冲

2.20 求出以下序列的自相关序列，证明它们都是偶序列。确定这两个自相关序列的最大值所在的位置。

(a) $x_1(n)=\alpha^n\mu(n)$

(b) $x_2(n)=\begin{cases}1 & 0\leqslant n\leqslant N-1\\ 0 & n=\text{其他}\end{cases}$

第 3 章 离散序列的两种傅里叶变换

本章研究离散时间信号的频域特性。对于连续信号，傅里叶变换和傅里叶级数是研究其频域特性的主要手段。用傅里叶变换可以求出连续信号的连续频谱；用傅里叶级数可以求出周期连续信号的离散频谱；对于离散序列，则存在着两种傅里叶变换——离散时间傅里叶变换(DTFT)和离散傅里叶变换(DFT)；前者(DTFT)用以求出离散信号的连续频谱，其定语“离散时间”意味着仅在时域离散，在频域是连续的；后者(DFT)用以求出连续频谱上的离散样本点，相当于对连续频谱进行等间隔采样，所以在时域和频域都是离散的，离散傅里叶变换就由此得名。请读者先把这两个名词加以区分，然后再仔细研究它的内涵。

3.1 序列信号的周期性和频率

3.1.1 序列信号的周期性

先回顾一下连续信号的周期性的概念。满足

$$x(t) = x(t+mP) \qquad m = 0, \pm 1, \pm 2, \cdots \tag{3.1.1}$$

的连续信号称为周期信号。一般说来，会有很多 P 满足此式，其中最小的非零 P 值称为信号的周期。对于正弦信号 $\sin \Omega_0 t$, 取 m=1，则周期性条件由下式确定。

$$\sin \Omega_0 t = \sin(\Omega_0 t + \Omega_0 P) = \sin(2\pi f_0 t + 2\pi f_0 P)$$

满足这个等式的条件为 $2\pi f_0 P = 2k\pi$, 即：$\Omega_0 = 2\pi f_0 = 2\pi k / P$，$P = 2k\pi / \Omega_0$。在 k=1 时，P 有最小值

$$P = 2\pi / \Omega_0 = 1 / f_0 \tag{3.1.2}$$

这就是正弦连续信号的周期，它等于以赫兹(Hz)为单位的频率的倒数。

在 2.3.1 节中，已经介绍过周期序列的概念。这里再把它与连续信号的周期性进行对比，得出更严格的定义。对于序列 $x(n)$，如果对所有的 n 满足

$$x(n) = x(n+N) \tag{3.1.3}$$

则这样的序列称为周期序列，满足此式的最小值 N 称为序列的周期。

按照这个定义，分析一个由模拟正弦信号 $x_a(t) = \sin(\Omega_0 t)$ 采样而得的正弦序列 $x(n)$=$\sin(\Omega_0 nT)$, 其中 T 为采样周期，看看它是否也是周期的。此时周期性条件（3.1.3）成为

$$\sin \Omega_0 nT = \sin \Omega_0 (n+N)T = \sin \Omega_0 nT \cos \Omega_0 NT + \cos \Omega_0 n \sin \Omega_0 NT \tag{3.1.4}$$

要使（3.1.4）式对所有的 n 成立，只有在 $\cos \Omega_0 NT = 1$ 和 $\sin \Omega_0 NT = 0$ 即 $\Omega_0 TN = 2k\pi$ 情况下，因而

$$\Omega_0 T = (2k / N)\pi \tag{3.1.5}$$

时成立。因为 k 和 N 都是整数，故 $2k/N$ 是一个有理数。可见，这要求 $\Omega_0 T$ 与 π 呈整数比数关系。

令 F_s=$1/T$ 为采样频率。模拟正弦信号的频率为 f_0=$\Omega_0/2\pi$，则有 f_0 / F_s=k/N。因此由模拟正

弦信号采样而得的正弦序列的周期性条件也可表为：模拟正弦信号的频率 f_0 与采样频率 F_s（均以 Hz 为单位）之比为两个整数之比。

例如，设 $\Omega_0 T=0.1\pi$，可求得 $N=2k\pi/0.1\pi=20k$，N 的最小值出现在 $k=1$ 时，即 $N=20$。若把 Ω_0 增大三倍，设 $\Omega_0 T=0.3\pi$，可求得 $N=2k\pi/0.3\pi=20k/3$，N 的最小整数值出现在 $k=3$ 时，仍有 $N=20$。两个波形图见图 3.1.1。运行 fgp311 演示程序可以得到图 3.1.1。

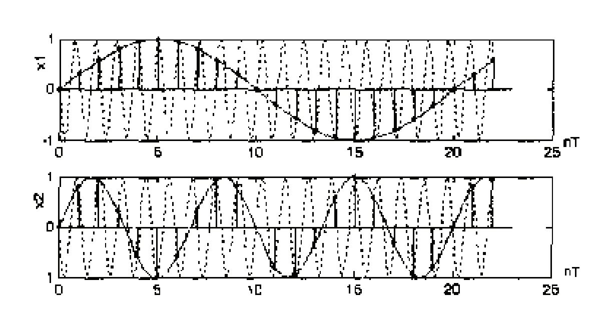


图 3.1.1　正弦序列信号的周期性和包络所含的频率成分

图 3.1.1 的连续波形表示模拟信号，而圆点表示采样序列。上图的模拟信号周期确实是下半图的三倍，但采样序列的周期却都是 20。因为下图的三个模拟周期中，采样点的位置不同，不满足序列周期性的要求（3.1.3）式。只有三个模拟周期以后，采样点的位置才会重复地出现。但人们的直觉是下半图中的序列 $x2$ 携带了较高频率的信号。于是就提出了一个问题：采样序列所携带的周期信号频谱，该怎样体现和定义？（图中虚线正弦波形的意义将在后面说明。）

3.1.2　序列信号的频率

先看连续余弦信号的频谱，分析频谱时最好采用复数指数形式，因而，把它写成

$$\cos\Omega_0 t=\frac{1}{2}\left(e^{j\Omega_0 t}+e^{-j\Omega_0 t}\right)$$

它由正、负两个 Ω_0 频率分量合成。在复平面上，正的 Ω_0 对应于反时针旋转的向量，负的 Ω_0 对应于顺时针旋转的向量，当这两个向量的幅度相同，而相角符号相反时，就合成为一个在实轴上的向量。它的相角为零，大小按余弦变化，形成了实信号 $\cos\Omega_0 t$，如图 3.1.2 所示。推而广之，任何实周期信号必然具有正、负两组频谱成分，两部分频谱的幅度对称而相位反对称，或者说，两部分频谱的实部对称而虚部反对称。频率与周期 P 的关系为 $\Omega_0=2\pi/P$，频率 Ω_0 的取值范围没有限制，即可以为（$-\infty$，∞）。

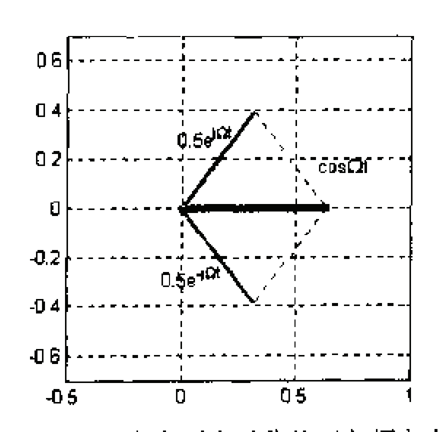


图 3.1.2　实序列由对称的正负频率合成

对虚信号 $j*\sin\Omega_0 t$ 可以做类似的分析，如下式。

$$j\cdot\sin\Omega_0 t=\frac{1}{2}\left(e^{j\Omega_0 t}-e^{-j\Omega_0 t}\right)$$

可见任何虚周期信号也必然具有正、负两个频率的频谱成分，只是要把负频率的频谱向量反号(这意味着使其幅度反号)后与正频率的频谱成分相加。这时两个向量的幅度相等而相位互补，或者说，正、负两部分频谱的实部反号而虚部同号。这时的合成向量在虚轴上，成为一个虚信号。运行 fgp312 演示程序可以看到图 3.1.2 的动画形式，它可分别演示余弦和正弦信号两种情况。

下面看正余弦序列的频率，如果采用与连续信号相同的定义，$f=1/P=1/(NT)$，由于图 3.1.1 中的两个波形具有同样的 N，就会得出两者的频率相同的结论。这显然与人们的直觉相违背。在图上，如果观察序列的包络，前一个信号在 20 个采样点中变化了一个周期，而后一个信号

变化了三个周期，因此在直观上应该用其包络的频率来定义序列的频率。包络是一根连续曲线，所以可以用连续信号的频率概念来定义它。上例中若 T=0.1s，则对第一个模拟波形，$\Omega_0=\pi$，对第二个波形，$\Omega_0=3\pi$。周期 P 分别为 2 和 2/3s。

序列 $\sin\Omega_0 nT$ 的包络是这样一个连续正弦信号 $\sin\Omega_1 t$，它在所有的采样点 $t=nT$ 处的值，都等于序列 $\sin\Omega_0 nT$ 的值。即 $\sin\Omega_1 t\big|_{t=nT}=\sin\Omega_0 nT$。满足这个条件的 Ω_1 实际上有无穷多个。因为对于任何整数 k 和 n，有

$$\sin\left(\Omega_0+\frac{2\pi k}{T}\right)t\Bigg|_{t=nT}=\sin(\Omega_0 nT+2\pi nk)=\sin\Omega_0 nT \tag{3.1.6}$$

这说明，如果 Ω_0 是一个序列的包络频率，则 $\Omega_1=\Omega_0+\dfrac{2\pi k}{T}=\Omega_0+2\pi kF_s\ (k=0,\pm1,\cdots)$ 都是它的包络频率。要得到惟一的答案，就必须加以限制。通常取所有 $|\Omega_0|$ 中的最小值，称为基本频率。因此序列正弦信号的包络的基本频率范围为：

$$-\frac{\pi}{T}\leqslant\Omega<\frac{\pi}{T} \tag{3.1.7}$$

也可表示为：

$$-0.5F_s\leqslant f<0.5F_s \tag{3.1.8}$$

按照包络的概念可以确定序列信号所携带的模拟频率。但要注意它是多义的，增减 k 个采样频率 F_s 都满足包络频率的定义。图 3.1.1 的虚线就是把原来频率减掉一个采样频率 F_s 后生成的正弦波，它也通过所有的序列样本点，所以也符合包络的定义。本来只有单一频率的模拟信号，经过采样后包含了无穷多个频率分量，说明采样是一个非线性过程。要避免多义性，通常取其最低频率。因而序列所能携带的最大包络频率为 $\pm0.5F_s$，这就是著名的奈奎斯特(Nyquist)频率。

可以用一个实验说明序列频率和基本频率的物理概念。让一个圆盘以频率 f_0（单位 Hz）作反时针转动，圆盘端面靠近边缘处有一个红点。用一个频率为 F_s 的闪光灯对它进行照射，人们只能在闪光灯发光的瞬间看到红点。由于闪光的间隔 $T=1/F_s$，两次闪光之间圆盘转过的角度为 $2\pi f_0T=2\pi f_0/F_s$。如果这个角度小于半圈，即 $2\pi f_0/F_s<\pi$，或 $f_0<0.5F_s$，人们可以感觉到圆盘上的红点在反时针转动；如果这个角度大于半圈而小于整圈，即 $\pi<2\pi f_0/F_s<2\pi$，或 $0.5F_s<f_0<F_s$，相当于顺时针看这个角度小于半圈，人们将以为圆盘上的红点在作顺时针转动，因为人眼具有 0.1s 左右的视觉暂留作用，他只能看到较低频率的动态过程，具有自然低通滤波的作用。假如 $f_0=0.7F_s$，$f_0-F_s=-0.3F_s$，也就是它等同于按负频率 $-0.3F_s$ 转动所得的采样序列，观察者会认为圆盘在顺时针转动。推而广之，圆盘以 $f_0=1.7F_s$, $2.7F_s$, … (反时针)或 $1.3F_s$, $-2.3F_s$, …(顺时针)转动时，采样出的样本都是同样的。因此，从采样的序列就无法惟一地判定原来的转速，除非指明要找的是它的最低的基本频率。图 3.1.3 表明了这几种情况。编者提供的 MATLAB 演示程序 fgp313 可在计算机上演示它的动画。

$\omega_0=\Omega_0T$ 称为序列信号的数字频率，它的单位是 rad/s，即是无量纲的。从图 3.1.3 中也可以看出它的物理意义。它是相邻两个采样时刻间圆盘转过的角度，以弧度 rad 为单位。考虑到正反转动的两种可能，规定只取绝对值小于等于 π 的角度，所以数字频率 ω 的主值取值范围为

$$-\pi\leqslant\omega<\pi \tag{3.1.9}$$

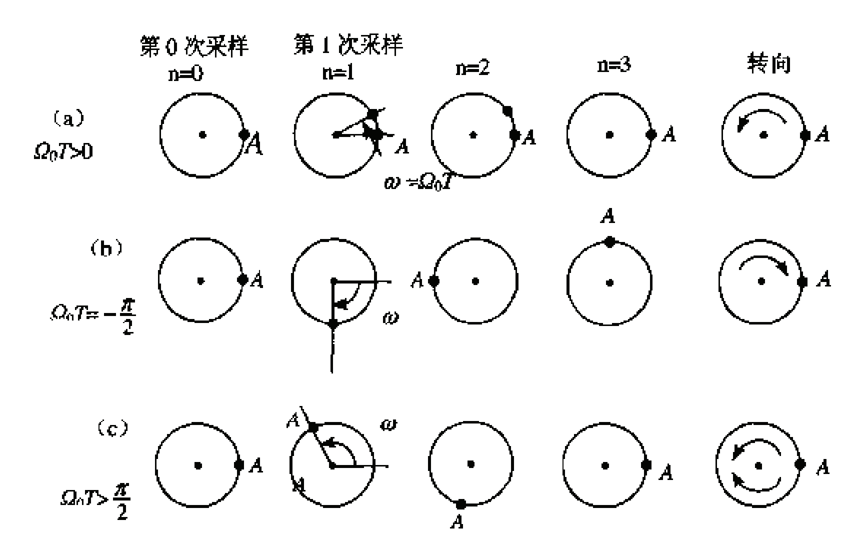


图 3.1.3　数字频率的物理意义示意图

不管原来的转动速度（频率 Ω_0）多高，经过 T 间隔采样后，相邻样本之间的转角只能取值在（$-\pi$，π）范围之内，因而反映的数字频率 $\omega=\Omega_0 T$ 也只能取值在这个范围。模拟频率和经采样后的数字频率的关系如图 3.1.4 所示，这是一个多值对单值的映射关系。一个模拟频率只能映射为一个数字频率，而一个数字频率却可以映射为无数个模拟频率。模拟频率可以从$-\infty$到∞，而数字频率只能取$-\pi$ 到 π。所以在离散序列中，确实包含着奈奎斯特频率范围之外的模拟频谱分量。这就是采样定理的核心概念，本书将在第 4 章中进行严格的证明和详细的讨论。

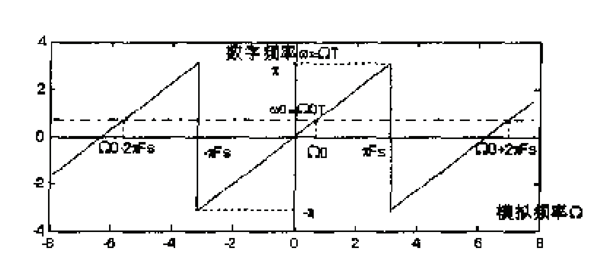


图 3.1.4　模拟频率和数字频率的关系

通常情况下，系统中都用低通滤波器把高于奈奎斯特频率的频谱过滤掉了，很少予以关心。不过现在也有一些新技术，专门利用高于奈奎斯特频率的频谱。例如，软件无线电中采用低采样率来传输高于奈奎斯特频率的信号。不过在本书中，只研究数字频率 ω 的主值范围内的频谱。

3.1.3　连续信号的频谱

上面讨论了带有单个频率分量的连续信号和离散序列，它的概念可以推广到有任意多个频率分量的情况。

周期连续信号的频率分量用傅里叶级数来表示，其傅里叶级数对可表为：

$$X_k=\frac{1}{P}\int_{-P/2}^{P/2}x(t)\mathrm{e}^{-\mathrm{j}k\Omega_0 t}\mathrm{d}t \qquad k=0,\ \pm1,\ \pm2,\ \cdots \tag{3.1.10}$$

$$x(t)=\sum_{k=-\infty}^{\infty}X_k\mathrm{e}^{\mathrm{j}k\Omega_0 t} \qquad X_k=|X_k|\mathrm{e}^{\mathrm{j}\varphi_k} \tag{3.1.11}$$

其中：P 为信号周期，$\Omega_0=2\pi/P$ 为信号的基波频率；k 为谐波的阶次；X_k是集中在基本频率的各阶倍频 $k\Omega_0$ 上的复数分量。可以看出，连续周期信号的频谱是离散的。

非周期连续信号的傅里叶变换对可以在(3.1.10)～(3.1.11)中，让 P 无限加大，或 $\Omega_0=2\pi/P$ 无限减小而得到：

$$X(\mathrm{j}\Omega)=\lim_{P\to\infty} X_m P=\int_{-\infty}^{\infty} x(t)\mathrm{e}^{-\mathrm{j}\Omega t}\mathrm{d}t \tag{3.1.12}$$

$$x(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty} X(\mathrm{j}\Omega)\mathrm{e}^{\mathrm{j}\Omega t}\mathrm{d}\Omega \tag{3.1.13}$$

此时 $\Omega=k\Omega_0$，在 $k\to\infty$和 $\Omega_0\to 0$ 时成为连续的模拟频率。而傅里叶级数变成了傅里叶变换。由于信号延伸至无限长，这种广义积分的成立是有条件的。它要求 $x(t)$满足绝对可积条件：

$$\int_{-\infty}^{\infty}|x(t)|\mathrm{d}t <\infty \tag{3.1.14}$$

理论上可以设计出不满足这个条件的信号。对于实际的信号，既没有无穷大的幅度，时间上也不可能无限延续，所以这个条件都能满足。可见，非周期连续信号的频率分量都可用傅里叶变换求得。它是分布在频率轴上的连续频谱。

应该认为，非周期连续信号是普遍的概念，从统一的观点出发，周期连续信号只是它的特例。谱是一种分布密度的概念，它的作用量应该和它在频谱图上所围的面积成正比。如果把频谱轴分成很多有限长的增量单元 $\Delta\Omega$，则（3.1.13）式中的积分变成了求和，即

$$x(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}\left[X(\mathrm{j}\Omega)\mathrm{d}\Omega\right]\mathrm{e}^{\mathrm{j}\Omega t} \;\to\; \frac{1}{2\pi}\sum_{\Omega-\infty}^{\infty}X(\mathrm{j}\Omega)\mathrm{e}^{\mathrm{j}\Omega t}\Delta\Omega$$

所以周期连续信号的傅里叶级数可以看成在 $\Omega=k\Omega_0$ 的频率点附近，集中了幅度极大的频谱，它与其邻域频率宽度的乘积 $X(\mathrm{j}\Omega)\ \Delta\Omega=X(\mathrm{j}k\Omega_0)\ \Delta\Omega$[或积分的面积 $\int_{\Omega-\Delta\Omega/2}^{\Omega+\Delta\Omega/2}X(\mathrm{j}k\Omega_0)\mathrm{d}\Omega$]等于其傅里叶级数 X_k。也就是说，这些傅里叶系数在频谱曲线上表现为所围面积与 X_k相当的脉冲函数 $\delta(\Omega-k\Omega_0)$。用这样的观点就可以把两者统一起来：傅里叶系数是对傅里叶变换（频谱）的采样，而傅里叶变换则可看成对傅里叶系数的拟合插补。因为一个是密度，另一个则是密度对频率的积分，两者之间差一个频率（或时间）量纲。后面会证明，所差的乘子恰恰是采样频率 $\Omega_s=2\pi F_s$，这和时域对连续信号采样的概念是一致的。采样序列与原模拟信号的量纲之间差一个时间乘子 T，傅里叶级数系数与连续频谱的量纲之间差一个频率乘子 Ω_s。

尽管在解析上使用无穷大、无限小等极限的概念，进而严格地划分了连续和离散、周期和非周期等信号类型和傅里叶方法。但一旦要用计算机进行实际计算时，必然要把连续信号离散化，也必然要把无限长信号有限化。不可能有真正的无穷大、无限小，因而也没有严格的连续和离散的界限。实际信号中任何一点微小的随机干扰就足以破坏它的周期性，所以也没有严格的周期和非周期的界限。这是学习本课程时应当具备的工程观点，也是本课程与“信号与系统”课程的一大区别。在本课程中研究频谱的分析计算，就要力图开发出适用于任何信号的算法或程序。

3.1.4 有限带宽序列

如果一个实序列在正频率区间的频谱占有 $0\leqslant\omega<\pi$ 的频带，（根据对称性，这等价于占有$-\pi\leqslant\omega<\pi$ 的全部频带，）称为全带宽序列。如果它只占上述频带中的一部分，就称为有限带宽序列。当一个序列信号在正频率区间所占的频带为 $0\leqslant\omega\leqslant\omega_p<\pi$，其中 ω_p 称为信号带宽，就称它为低通信号序列；若它所占的频带为 $0\leqslant\omega_L\leqslant\omega_H\leqslant\omega<\pi$，就称它为带通信号序列，$\omega_H-\omega_L$ 称为它的带宽。

可以证明，带宽有限的序列在时域必然具有无限长度；反过来，时域长度有限的序列在

频域的带宽必然是无限的。所以，不可能存在时域长度和频域带宽都有限的信号和序列。

3.2　离散时间傅里叶变换(DTFT)

3.2.1　离散时间傅里叶变换(DTFT)的定义

对于序列 $x(n)=x(nT)$，其离散时间傅里叶变换的定义为：

$$X(\mathrm{j}\Omega)=F[x(nT)]=\sum_{n=-\infty}^{\infty}x(nT)\mathrm{e}^{-\mathrm{j}\Omega nT_s}\qquad -\infty\leqslant\Omega\leqslant\infty \tag{3.2.1 a}$$

用数字频率来表示，可以写成：

$$X(\mathrm{j}\omega)=F[x(n)]=\sum_{n=-\infty}^{\infty}x(n)\mathrm{e}^{-\mathrm{j}\omega n}\qquad -\infty\leqslant\omega\leqslant\infty \tag{3.2.1 b}$$

离散时间傅里叶变换收敛的充分必要条件是 $x(n)$绝对可加，即

$$\sum_{-\infty}^{\infty}|x(n)|<\infty$$

可以这样认为：序列中的每一个样本 $x(n)$对频谱产生的贡献为 $x(n)\mathrm{e}^{-\mathrm{j}\Omega nT}$，其中 $x(n)$是它的幅度，而$-\Omega Tn=-\omega\cdot n$ 是它的相位。它既取决于频率 ω，又取决于样本在时间轴上的位置 n。把整个序列中所有样本的频谱分量按向量（复数）算法叠加起来，就得到序列的频谱 $X(\mathrm{j}\Omega)$。公式(3.2.1a,b)就是离散时间傅里叶变换，英文缩写为 DTFT (Discrete-Time Fourier Transform)。它的一个对应物是连续时间傅里叶变换 CTFT (Continuous-Time Fourier Transform)。它的另一个对应物是离散傅里叶变换，英文缩写为 DFT (Discrete Fourier Transform)。它与前一个对应物的差别在于时域中离散与连续之分，它与后一个对应物的差别在于频域中连续与离散之分。离散时间傅里叶变换的起名缘由，在于强调它只在时间域离散，而频谱是连续的。而离散傅里叶变换则在时域和频域都是离散的。许多中文书中称 DTFT 为“序列的傅里叶变换”，这个中文术语不好，因为它没有和序列的离散傅里叶变换准确区分。因此，本书采用其英文的直译名。

根据定义，$X(\mathrm{j}\omega)$一般应该为复数，可以分解为实部 $X_{\mathrm{re}}(\mathrm{j}\omega)$ 和虚部 $\mathrm{j}X_{\mathrm{im}}(\mathrm{j}\omega)$：

$$X(\mathrm{j}\omega)=X_{\mathrm{re}}(\mathrm{j}\omega)+\mathrm{j}X_{\mathrm{im}}(\mathrm{j}\omega)$$

由于 $\mathrm{e}^{\mathrm{j}\omega n}=\cos\omega n+\mathrm{j}\sin\omega n$，将它代入(3.2.1 b)后不难看出，实部和虚部分别对应于：

$$X_{\mathrm{re}}(\mathrm{j}\omega)=\sum_{-\infty}^{\infty}x(n)\cos\omega n$$

$$X_{\mathrm{im}}(\mathrm{j}\omega)=\sum_{-\infty}^{\infty}x(n)\sin\omega n$$

由 $\cos\omega n$ 为偶函数，$\sin\omega n$ 为奇函数，不难证明，$X_{\mathrm{re}}(\mathrm{j}\omega)=X_{\mathrm{re}}(-\mathrm{j}\omega)$ 及 $X_{\mathrm{im}}(\mathrm{j}\omega)=-X_{\mathrm{im}}(-\mathrm{j}\omega)$。由此可知，实数序列的 DTFT 的实部为偶对称，虚部为奇对称。也可以把 $X(\mathrm{j}\,\omega)$写成幅度和相角的复数形式：

$$X(\mathrm{j}\omega)=|X(\mathrm{j}\omega)|\mathrm{e}^{\mathrm{j}\theta(\omega)}$$

很容易得知，其幅值的平方 $|X(\mathrm{j}\omega)|^2=X_{\mathrm{re}}{}^2(\mathrm{j}\omega)+X_{\mathrm{im}}{}^2(\mathrm{j}\omega)$ 是偶函数，其相位函数

$\theta(\omega)=\arg[X(j\omega)] = \arctan[X_{im}(j\omega)/X_{re}(j\omega)]$是奇函数，和实模拟信号的傅里叶变换有同样的结论。其实，这个结论在第 3.1 节图 3.1.2 中已经得到了，这里只是在更普遍意义下的一个数学证明。

3.2.2 DTFT 的性质

DTFT 是傅里叶变换（FT）中的一种，因而它具备傅里叶变换的全部性质，另外再加上自己的一些特性。许多性质可以很直观地得到，读者也可以自己证明，这里就只列出结论。

（1）时域离散频域连续性

DTFT 的第一个特性就是，它在时间轴上是离散取值，而在频率域上则连续取值。故它的正变换[见(3.2.1)式]是离散求和的形式，而离散时间傅里叶反变换（IDTFT）却表现为积分形式：

$$x(n)=F^{-1}[X(j\Omega)]=\frac{T}{2\pi}\int_{-\pi/T_s}^{\pi/T_s}X(j\Omega)e^{j\Omega Tn}d\Omega \tag{3.2.2 a}$$

用数字频率 ω=TΩ 积分表示的反变换表示式为

$$x(n)=F^{-1}[X(j\omega)]=\frac{1}{2\pi}\int_{-\pi}^{\pi}X(j\omega)e^{j\omega n}d\omega \tag{3.2.2 b}$$

DTFT 正变换是求和式（3.2.2a），而反变换 IDTFT 是积分式（3.2.2b）。这非常明确地反映了它在时域离散取值，而在频域连续取值的特征。

（2）周期性

根据 DTFT 的定义，考虑到$e^{-2\pi Mn}=1$(M 与 n 为整数时)，有

$$X(j\omega)=F[x(n)]=\sum_{-\infty}^{\infty}x(n)e^{-j\omega n}=\sum_{-\infty}^{\infty}x(n)e^{-j(\omega+2\pi M)n}=X(j(\omega+2\pi M)) \tag{3.2.3}$$

可见 $X(j\omega)$以 $\omega=2\pi$ 为周期。这正是前面用物理概念探讨的序列频谱具有周期性的问题。由此只需要知道 $X(j\omega)$的一个周期，即 $\omega=[0,2\pi]$,或$[-\pi,\pi]$，就可以分析序列的频谱，不需要取整个 $-\infty<\omega<\infty$域来分析。

DTFT 与其他所有的傅里叶变换所具有的共性见表 3.6.2。

（3）线性

设$G(e^{j\omega})=F[g(n)]$，$H(e^{j\omega})=F[h(n)]$，则

$$F[ag(n)+bh(n)]=aG(e^{j\omega})+bH(e^{j\omega}) \tag{3.2.4}$$

式中 a，b 为常数。

（4）时移性

$$F[x(n-n_0)]=e^{-j\omega n_0}X(e^{j\omega}) \tag{3.2.5}$$

（5）频移性

$$F[e^{j\omega_0 n}x(n)]=X(e^{j(\omega-\omega_0)}) \tag{3.2.6}$$

（6）时域导数

$$F[n\cdot g(n)]=j\frac{dG(e^{j\omega})}{d\omega} \tag{3.2.7}$$

（7）时域卷积定理

设在时域 $y(n)$是 $x(n)$和 $h(n)$的卷积 $y(n)=x(n)\otimes h(n)$，则 $y(n)$的 DTFT 是 $x(n)$和 $h(n)$的

DTFT 的乘积

$$Y(e^{j\omega}) = X(e^{j\omega}) \cdot H(e^{j\omega}) \tag{3.2.8}$$

证明：$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m)$，$Y(e^{j\omega}) = F[y(n)] = \sum_{n=-\infty}^{\infty}\left[\sum_{m=-\infty}^{\infty} x(m)h(n-m)\right]e^{-j\omega n}$。

令 $k = n - m$，则

$$Y(e^{j\omega}) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} h(k)x(m)e^{-j\omega k}e^{-j\omega m} = \sum_{n=-\infty}^{\infty} h(k)e^{-j\omega k}\sum_{m=-\infty}^{\infty} x(m)e^{-j\omega m} = H(e^{j\omega}) \cdot X(e^{j\omega})$$

该定理说明，两序列卷积的 DTFT 服从相乘的关系。对于线性时不变系统，输出的 DTFT 等于输入信号的 DTFT 乘以单位脉冲响应的 DTFT。因此，在求系统的输出信号时，可以在时域用卷积公式(2.3.7)计算，也可以在频域按照(3.2.8)式求出输出的 DTFT，再作 IDTFT，求出输出信号。

（8）频域卷积定理

设 $y(n) = x(n) \cdot h(n)$，则

$$Y(e^{j\omega}) = \frac{1}{2\pi} X(e^{j\omega}) \otimes H(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})H(e^{j(\omega-\theta)})d\theta \tag{3.2.9}$$

证明：$Y(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)h(n)e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x(n)\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\theta})e^{j\theta n}d\theta\right]e^{-j\omega n}$

交换积分与求和的次序，得到：

$$Y(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\theta})\left[\sum_{n=-\infty}^{\infty} x(n)e^{-j(\omega-\theta)n}\right]d\theta$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\theta})X(e^{j(\omega-\theta)})d\theta = \frac{1}{2\pi} X(e^{j\omega}) \otimes H(e^{j\omega})$$

该定理表明，在时域两序列相乘，转换到频域后，它们服从卷积关系。

（9）帕瑟瓦尔（Parseval）定理

$$\sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |x(e^{j\omega})|^2 d\omega \tag{3.2.10}$$

证明：

$$\sum_{n=-\infty}^{\infty} |x(n)|^2 = \sum_{n=-\infty}^{\infty} x(n)x^*(n) = \sum_{n=-\infty}^{\infty} x^*(n)\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega\right]$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})\sum_{n=-\infty}^{\infty} x^*(n)e^{j\omega n}d\omega$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})X^*(e^{j\omega})d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(e^{j\omega})|^2 d\omega$$

帕瑟瓦尔定理说明，序列信号时域的总能量等于频域的总能量。这里频域总能量是指 $|X(e^{j\omega})|^2$ 在一个周期中的积分再乘以 $1/(2\pi)$。正如第 2 章已经指出那样，这里所说的序列的能量和功率只是与连续域能量和功率计算形式相仿的运算，并不是物理上的能量和功率。

以上这些 DTFT 的性质归纳于表 3.6.2 中备查。

3.2.3　实数奇偶序列的 DTFT

根据(2.3.2)式，实数序列 $x(n)$可以分解成偶序列和奇序列以下两个分量。

$$x(n)= x_e(n)+ x_o(n)$$

其中偶部和奇部分别由下式给出：

$$x_e(n)= [x(n)+x(-n)]/2;\quad x_o(n)= [x(n)-x(-n)]/2$$

可以证明，实数序列 $x(n)$ 分解出的偶序列 $x_e(n)$ 恰好对应于其 DTFT 的实部 $X_{re}(j\omega)$。它分解出的奇序列 $x_o(n)$对应于其 DTFT 的虚部 $X_{im}(j\omega)$。

证明：先对偶序列等式两端进行 DTFT 变换，注意

$$F[x(n)]=\sum_{n=-\infty}^{\infty}x[n](\cos\omega n+\mathrm{j}\sin\omega n)=X_{re}(\mathrm{j}\omega)+\mathrm{j}X_{im}(\mathrm{j}\omega)$$

$$F[x(-n)]=\sum_{n=-\infty}^{\infty}x[n](\cos\omega n-\mathrm{j}\sin\omega n)=X_{re}(\mathrm{j}\omega)-\mathrm{j}X_{im}(\mathrm{j}\omega)$$

$$\begin{aligned}F[x_e(n)]&=\{F[x(n)]+F[x(-n)]\}/2\\&=[X_{re}(\mathrm{j}\omega)+\mathrm{j}X_{im}(\mathrm{j}\omega)+X_{re}(\mathrm{j}\omega)-\mathrm{j}X_{im}(\mathrm{j}\omega)]/2=X_{re}(\mathrm{j}\omega)\end{aligned}\tag{3.2.11}$$

用同样的方法可以证明，奇序列 $x_o(n)$对应于 DTFT 的虚部 $X_{im}(j\omega)$。

3.2.4　复数奇偶序列的 DTFT

同样可以证明，复数序列 $x(n)$分解出的偶序列 $x_e(n)$恰好对应于其 DTFT 的实部 $X_{re}(j\omega)$。它分解出的奇序列 $x_o(n)$则对应于其 DTFT 的虚部 $X_{im}(j\omega)$。

证明：任意复数信号序列 $x(n)$可以用它的实部和虚部表示如下。

$$x(n) = x_{re}(n) + \mathrm{j}x_{im}(n)\tag{3.2.12}$$

它的 DTFT 可以展开为：

$$F[x(n)]=\sum_{n=-\infty}^{\infty}[x_{re}(n)+\mathrm{j}x_{im}(n)](\cos\omega n+\mathrm{j}\sin\omega n)$$

$$=\sum_{n=-\infty}^{\infty}[x_{re}(n)\cos\omega n+x_{im}(n)\sin\omega n]+\mathrm{j}\sum_{n=-\infty}^{\infty}[x_{im}(n)\cos\omega n+x_{re}(n)\sin\omega n]$$

得到

$$F[x(n)]=X_{re}(\mathrm{j}\omega)+\mathrm{j}X_{im}(\mathrm{j}\omega)=X(\mathrm{j}\omega)\tag{3.2.13}$$

其中 $X_{re}(j\omega)$和$X_{im}(j\omega)$ 分别为 $F[x(n)]=X(j\omega)$ 的实部和虚部。

由第 2.3.3 节或知，$x(n)$又可分解成它的偶部和奇部：

$$x(n)=x_e(n)+x_o(n)\tag{2.3.5}$$

其中偶部和奇部都是复数序列，并分别由下式给出：

$$x_e(n)=\frac{1}{2}[x(n)+x^*(-n)]\tag{2.3.6 a}$$

$$x_o(n)=\frac{1}{2}[x(n)-x^*(-n)]\tag{2.3.6 b}$$

现求 $x^*(-n)$ 的 DTFT，注意与推导(3.2.13)式的过程对照，得到

$$F[x^*(-n)]=\sum_{n=-\infty}^{\infty}[x_{re}(n)-jx_{im}(n)](\cos\omega n-j\sin\omega n)$$
$$=\sum_{n=-\infty}^{\infty}[x_{re}(n)\cos\omega n+x_{im}(n)\sin\omega n]-j\sum_{n=-\infty}^{\infty}[x_{im}(n)\cos\omega n+x_{re}(n)\sin\omega n]$$
$$=X_{re}(j\omega)-jX_{im}(j\omega)$$

对(2.3.6 a)式求 DTFT，得到

$$\begin{aligned}F[x_e(n)]&=\{F[x(n)]+F[x^*(-n)]\}/2\\&=[X_{re}(j\omega)+jX_{im}(j\omega)+X_{re}(j\omega)-jX_{im}(j\omega)]/2=X_{re}(j\omega)\end{aligned}\tag{3.2.14}$$

同样，对(2.3.6 b)式求 DTFT，就可以得到 $F[x_o(n)]=\{F[x(n)]-F[x^*(-n)]\}/2=X_{im}(j\omega)$，这就证明了本命题。

由此可见，复数序列完全可以用实数序列的结论。复数序列的偶序列也就是共轭对称序列 $x_e(n)$。它恰好对应于其 DTFT 的实部 $X_{re}(j\omega)$。它分解出的奇序列，即共轭反对称序列 $x_o(n)$，对应于其 DTFT 的虚部 $X_{im}(j\omega)$。

3.2.5　DTFT 的解析计算示例

目前已经求出了一些基本序列的傅里叶变换，现列于表 3.2.1 中，以便用查表的方法求 DTFT 和 IDTFT。

表 3.2.1　基本序列的离散时间傅里叶变换

序列(IDTFT)	离散时间傅里叶变换(DTFT)
单个脉冲 $\delta(n)$	1（频域的等幅脉冲串）
$a^n\mu(n),\|a\|<1$	$(1-ae^{-j\omega})^{-1}$
方波序列 $R_N(n)$	$e^{-j(N-1)\omega/2}\sin(\omega N/2)/\sin(\omega/2)$
单位阶跃序列 $\mu(n)$	$(1-e^{-j\omega})^{-1}+\sum_{k=-\infty}^{\infty}\pi\delta(\omega-2\pi k)$
周期脉冲序列 $x(n)=1$，$[n\in(-\infty,\ \infty)]$	$2\pi\sum_{k=-\infty}^{\infty}\delta(\omega-2\pi k)$
$e^{j\omega_0 n}$, $2\pi/\omega_0$ 有理数	$2\pi\sum_{k=-\infty}^{\infty}\delta(\omega-\omega_0-2\pi k)$
$\cos\omega_0 n$, $2\pi/\omega_0$ 有理数	$\pi\sum_{k=-\infty}^{\infty}[\delta(\omega-\omega_0-2\pi k)+\delta(\omega+\omega_0-2\pi k)]$
$\sin\omega_0 n$, π/ω_0 有理数	$-j\pi\sum_{k=-\infty}^{\infty}[\delta(\omega-\omega_0-2\pi k)-\delta(\omega+\omega_0-2\pi k)]$

例 3.2.1：以表中倒数第二项单位阶跃序列 $\mu(n)$为例，求它的 DTFT。

解：为了充分利用已有的结果，把单位阶跃序列 $\mu(n)$和单位脉冲序列 $\delta(n)$之间的差分关系建立如下。

$$\mu(n)-\mu(n-1)=\delta(n)\tag{3.2.15}$$

又根据它与它的移位折叠相加成为周期脉冲序列 $x(n)=1$，其中，1 表示在 n 在$(-\infty,\ \infty)$范围内的全部

序列值 $x(nT)$均等于 1，也就是周期为 T、幅度为 1 的等幅脉冲串。

$$\mu(n)+\mu(-n+1)=x(n)=1 \tag{3.2.16}$$

对此两式进行 DTFT，由表 3.2.1 查得 1 的 DTFT 是在 ω轴上原点（即直流）的一个加权值为 $\dfrac{2\pi}{T}$ 的脉冲，并按 2π 为周期向外延拓

$$\text{DTFT}\left[1\right]=\frac{2\pi}{T}\delta(\Omega)=2\pi\sum_{k=-\infty}^{\infty}\delta(\omega-2\pi k) \tag{3.2.17}$$

再利用时域移位性质，得到：

$$U(\mathrm{e}^{\mathrm{j}\omega})-\mathrm{e}^{\mathrm{j}\omega}U(\mathrm{e}^{\mathrm{j}\omega})=1 \tag{3.2.18}$$

$$U(\mathrm{e}^{\mathrm{j}\omega})+\mathrm{e}^{\mathrm{j}\omega}U(\mathrm{e}^{-\mathrm{j}\omega})=2\pi\sum_{k=-\infty}^{\infty}\delta(\omega-2\pi k) \tag{3.2.19}$$

由(3.2.18)式求得

$$U(\mathrm{e}^{\mathrm{j}\omega})=\frac{1}{1-\mathrm{e}^{-\mathrm{j}\omega}}+\mathrm{C} \tag{3.2.20}$$

其中 C 是差分方程求解的任意积分常数。正如连续系统微分方程积分后有积分常数一样。其实为了求出这个积分常数，必须要用边界条件。将(3.2.20)代入(3.2.19)式，得到

$$C=-\pi\sum_{k=-\infty}^{\infty}\delta(\omega-2\pi\, k) \tag{3.2.21}$$

最后得

$$U(\mathrm{e}^{\mathrm{j}\omega})=\frac{1}{1-\mathrm{e}^{-\mathrm{j}\omega}}-\pi\sum_{k=-\infty}^{\infty}\delta(\omega-2\pi\, k) \tag{3.2.22}$$

例 3.2.2：以表中倒数第二项为例。令 $\tilde{x}(n)=\cos\omega_0 n,\ 2\pi/\omega_0$ 为有理数，求其 DTFT。

解：将 $\tilde{x}(n)$ 用欧拉公式展开：

$$\tilde{x}(n)=\frac{1}{2}\left[\mathrm{e}^{\mathrm{j}\omega_0 n}+\mathrm{e}^{-\mathrm{j}\omega_0 n}\right]$$

按照(3.2.8）式，其 DTFT 推导如下：

$$X(\mathrm{e}^{\mathrm{j}\omega})=\text{DTFT}[\cos\omega_0 n]=\frac{1}{2}\bullet 2\pi\sum_{r=-\infty}^{\infty}[\delta(\omega-\omega_0-2\pi\, k)+\delta(\omega+\omega_0-2\pi\, r\, k)]$$

$$=\pi\sum_{r=-\infty}^{\infty}[\delta(\omega-\omega_0-2\pi\, k)+\delta(\omega+\omega_0-2\pi\, k)] \tag{3.2.23}$$

上式表明，$\cos\omega_0 n$ 的 DTFT 是在 $\omega=\pm\omega_0$ 处的单位冲激函数，强度为 π，且以 2π 为周期进行延拓，如图 3.2.1 所示。

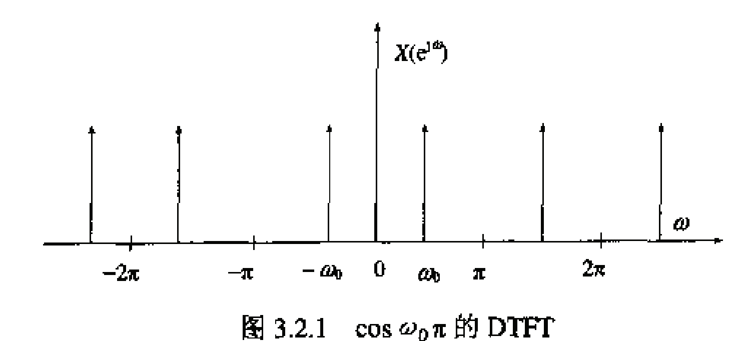


图 3.2.1　$\cos\omega_0\pi$ 的 DTFT

3.2.6 DTFT 的数值计算示例

例 3.2.3：考虑矩形窗函数序列

$$x(nT)=\begin{cases}1 & 对于|n|\leqslant M\\ 0 & 对于|n|\geqslant M\end{cases}$$

其中 T=0.5。该序列是长度为 N=2M+1 的双边序列。它的 DTFT 等于：

$$X(\Omega)=\sum_{n=-\infty}^{\infty}x(nT)\mathrm{e}^{-\mathrm{j}\Omega nT}=\sum_{n=-M}^{M}\mathrm{e}^{-\mathrm{j}\Omega nT}=\mathrm{e}^{\mathrm{j}\Omega MT}\sum_{n=0}^{2M}\left(\mathrm{e}^{-\mathrm{j}\Omega T}\right)^{n}$$

利用级数求和关系式：

$$\sum_{r=0}^{N-1}r^{n}=1+r^{2}+\cdots+r^{N-1}=\frac{1-r^{N}}{1-r} \tag{3.2.24}$$

可得

$$X(\Omega)=\mathrm{e}^{\mathrm{j}\Omega MT}\frac{1-\mathrm{e}^{-\mathrm{j}\Omega(2M+1)T}}{1-\mathrm{e}^{-\mathrm{j}\Omega T}}=\frac{\mathrm{e}^{\mathrm{j}\Omega MT}-\mathrm{e}^{-\mathrm{j}\Omega(M+1)T}}{1-\mathrm{e}^{-\mathrm{j}N\Omega T}}$$

$$=\frac{\mathrm{e}^{-\mathrm{j}0.5\Omega T}\left(e^{\mathrm{j}\Omega(M+0.5)T}-\mathrm{e}^{-\mathrm{j}\Omega(M+0.5)T}\right)}{\mathrm{e}^{-\mathrm{j}0.5\Omega T}\left(\mathrm{e}^{\mathrm{j}0.5\Omega T}-\mathrm{e}^{-\mathrm{j}0.5\Omega T}\right)} \tag{3.2.25}$$

$$=\frac{2\mathrm{j}\sin[(M+0.5)\Omega T]}{2\mathrm{j}\sin(0.5\Omega T)}=\frac{\sin(0.5N\Omega T)}{\sin(0.5\Omega T)}$$

所得频谱是实数。在 M=5，T=0.5 的情况下，用下列 MATLAB 语句来计算(3.2.25)式。其中有些参数要自己选择，比如：

（1）序列的长度，按照题意，x 是一个无限长的序列：除了-M 到 M 的区间，其他位置都为零。因此截取多长的一段可以由读者自行决定，但必须比 N 大很多。此处取 n=-16:16 的一段。

（2）频率范围和分辨率，前已指出，序列的频谱是周期性的，其基本频率范围在[$-\pi/T$，π/T]即[-6.28，6.28]范围内，这里故意选大一些，取为[-15，15]，以便看出频谱的周期性。

（3）可以看出 $\dfrac{\sin(0.5N\,\Omega\,T)}{\sin(0.5\,\Omega\,T)}$ 在 Ω=0 和 $\Omega=2k\pi/T$ 处会成为不定式 0/0，要求出这些点的值，有两个办法：一种办法是用罗必达法则，那就要分别求分子分母对 Ω 的导数，并求它们在 Ω=0 和 $2k\pi/T$ 处的值，这很不方便；高明的办法是在计算时避开这些奇异点，在设定频率向量 Ω 时，把各个频点的频率都加一个微小量（例如 MATLAB 中的 eps，有时嫌 eps 太小，取 10^{-10}，表为 1e-10），使 Ω 不取到零值。这样不必加任何其他语句就解决了问题。由此，画图的程序 hc323 中语句如下：

```
M=4; N=2*M+1; T=0.5; n= -16:16;
x=[zeros(1, 12),ones(1, N), zeros(1, 12)];
Omega=[-15:0.1:15]+1e-10;
X=sin(0.5*N*Omega*T)./sin(0.5*Omega*T);
```

以上是核心语句。注意 X 是复数，用 stem (n, x)和 plot (Omega, real(X))命令，可画出它的时间序列和频谱曲线如图 3.2.2 所示。

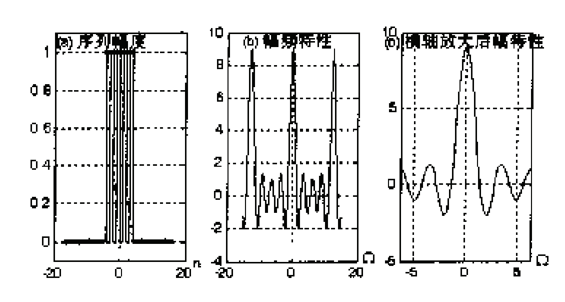


图 3.2.2　矩形序列的幅频特性

由于 (3.2.25) 式的零点位于$\Omega = k\dfrac{2\pi}{NT}$　$k = \pm 1, \pm 2, \cdots$处。k=[−1,+1]的区间称为主瓣，它最宽，占了$\dfrac{4\pi}{NT}$，其他为副瓣，共有 N−2 个，每个宽度为$\dfrac{2\pi}{NT}$。主瓣和副瓣加起来的范围为$-\dfrac{\pi}{T} \leqslant \Omega \leqslant \dfrac{\pi}{T}$，正好就是序列的奈奎斯特频率范围的大小，如图 3.2.2 右子图所示。

例 3.2.4：考虑长度为 4 的有限序列，$x(0)=2, x(1)=-1, x(2)=1$ 及 $x(3)=1$，求它的 DTFT。

解：按定义

$$X(\mathrm{j}\omega) = \sum_{-\infty}^{\infty} x(n)\mathrm{e}^{-\mathrm{j}\omega n} = 2 - \mathrm{e}^{-\mathrm{j}\omega} + \mathrm{e}^{-\mathrm{j}2\omega} + \mathrm{e}^{-\mathrm{j}3\omega} \tag{3.2.26}$$

因为没有给出采样周期，式中采用了数字频率 $\omega=\Omega T$。ω 的基频范围为[−π，π]。注意 $x(n)$ 在时域的序列是离散的，这里只有四个点。而它的 $X(\mathrm{j}\omega)$在频域则是连续的，即可以在[−∞，∞] 范围内取任意值。考虑了周期性，它的基本频率可以在[−π，π]范围内任意地连续取值，代入(3.2.26)式，即可求出一系列的 $X(\omega)$，因为 $X(\omega)$是复数，可以分解为幅度和相位，并画出幅度和相位随频率变化的曲线。虽然本题只给了四个点的序列，(3.2.26)式已非常简单，但要计算几十个频点的 $X(\omega)$，再把它们连起来形成连续曲线，也是很麻烦、费时且容易出错的工作。因此利用 MATLAB 来进行计算是顺理成章的做法。

3.2.7　用 MATLAB 计算 DTFT

用 MATLAB 来计算 DTFT 和 IDTFT 时的问题在于，它无法计算连续变量 ω（今后一般都用数字频率 ω），只能在$-\pi \leqslant \omega < \pi$（或考虑到对称性，在 $0 \leqslant \omega < \pi$）范围内，把 ω 赋值为很密的、长度很长的向量来近似连续变量。通常最简单的就是赋以 K 个等间隔的值。

$$\omega = k \cdot \mathrm{d}\omega = k\frac{2\pi}{K} \tag{3.2.27}$$

其中：$\mathrm{d}\,\omega=2\pi/K$，称为频率分辨率。它表示把基本数字频率的范围 2π 均分成 K 份后，每一份的大小，k 是表示频率序数的整数向量，比如 k=0:K−1，…等等。它的取值可以有几种方法：通常在 DTFT 中，频率取$-\pi \leqslant \omega < \pi$ 的范围，当 K 为偶数时，取

$$k = -K/2, -K/2+1, \cdots, -1, 0, 1, \cdots, K/2-1$$

如果 K 为奇数，则取

$$k = -K/2+0.5, \cdots, -1, 0, 1, \cdots, K/2-0.5$$

以数字例，若 K=8，则 k=[−4, −3, −2, −1, 0, 1, 2, 3]，如 K=7，则 k=[−3, −2, −1, 0, 1, 2, 3]。在 K 为奇数时，得出对称的 k，这比较好理解。当 K 为偶数时（这是绝大多数情况），得出的 k 却关于 ω=0 不对称，其原因在于 k=−4 和 4 在圆周上对应于同一点 ω=±π。对于周期性

的频谱，其中的一个点应该属于另一个周期，不能重复计算在同一周期内。在 MATLAB 信号处理工具箱中，就规定了 $-\pi \leqslant \omega < \pi$（范围，意思是左边为闭区间，右边为开区间，即奈奎斯特频率范围不包括 $\omega=+\pi$ 点而包括 $\omega=-\pi$ 点。为什么这样规定？这是因为有时只要取正半部奈奎斯特频率 $0 \leqslant \omega < \pi$。此时一定要把 $\omega=0$ 取为向量的起点的，因而必须甩掉 $\omega=\pi$ 点，把它当做下一个周期的起点。为了便于在正频段和正负全频段之间转换，就采取了前后一致的定义方法。

可以为奇偶两种情况综合出一个共同的确定频序向量 k 的公式，那就是

$$k = \underline{-(K-1)/2} : \underline{(K-1)/2} \tag{3.2.28a}$$

下画线表示向下取整。读者可按上述 K=8 和 7 两种情况检验此式的正确性。

在 MATLAB 中的向下取整函数为 floor，floor(x)的意义是把 x 向下（向 $-\infty$）取整，所以与(3.2.28）式等价的 MATLAB 语句为

k=floor((-K/2+0.5):(K/2-0.5))　　(3.2.28 b)

读者可以自行测试并解释其正确性。如果只取正半部奈奎斯特频率，则有：

k=floor(0:(K/2-0.5))　　(3.2.28 c)

还有些情况要取 $0 \leqslant \omega < 2\pi$ 范围的，则有

k=floor(0:(K-0.5))　　(3.2.28 d)

给定了输入序列（包括序列 x 及其位置向量 n），又设定了要求的频率分辨率 dω及频序向量 k，则 DTFT 的计算式(3.2.1 a)可以用一个向量与矩阵相乘的运算来实现。

$$[X(\omega_1), X(\omega_2), \cdots, X(\omega_K)] = [x(n_1), \cdots, x(n_N)] \cdot \begin{bmatrix} e^{j\omega_1 n_1} & e^{j\omega_2 n_1} & \cdots e^{j\omega_K n_1} \\ e^{j\omega_1 n_2} & e^{j\omega_2 n_2} & \cdots e^{j\omega_K n_2} \\ \cdots & \cdots & \vdots \\ e^{j\omega_1 n_N} & e^{j\omega_2 n_N} & \cdots e^{j\omega_K n_N} \end{bmatrix} \tag{3.2.29}$$

如果频率向量表为 $\omega=[\omega_1,\omega_2,\cdots,\omega_K]=k \cdot \mathrm{d}\omega$，而序列的位置向量为 n=[n1 : nN]，则(3.2.29)式中的矩阵的指数部分可以写成 $\mathrm{j}\, n^T * \omega = \mathrm{j}n^T * k\mathrm{d}\omega$,用 MATLAB 语句表达时，把 ω 代以 w,转置符号 n^T 换成 MATLAB 中的相应符号 n'，则求 DTFT 的程序可以写得非常简洁：

X=x*exp(j*dw*n'*k)　　(3.2.30a)

如果 x 和 X 都是列向量，也可以有另一种形式，这就要把后面的矩阵也转置一下，写成：

X= exp(j*dw*k'*n)*x　　(3.2.30b)

读者可把它写成矩阵形式，验证此式的正确性。现在用它来算例 3.2.4，假如取 64 个频点，核心语句为：

```
x=[2,-1,1,1];nx=0:3;K=64;dw=2*pi/K;
k=floor((-K/2+0.5):(K/2-0.5));
X=x*exp(j*dw*nx'*k);
```

然后就可用 plot(k*dw,abs(X)) 及 plot(k*dw,angle(X))语句画出其幅频和相频特性。如图 3.2.3 所示。读者可以在执行此程序后做分阶动作，把 K 取小些，如取 6，观察在 MATLAB 中执行了 nx'*k 和 exp(j*dw*nx'*k)分别得出的结果，以加深对这个简练程序的理解。

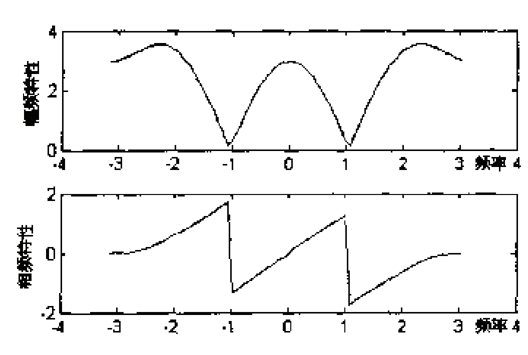


图 3.2.3　例 3.2.2 的幅频和相频特性

从得到的幅频和相频特性可以看到，它的幅频特性是对称的，即偶函数，而其相频特性则是奇函数。这符合实数离散时间傅里叶变换的特性。

MATLAB 信号处理工具箱中编有计算 DTFT 的专用函数 freqz。它的调用方式为：

[H,w] = freqz (b,a,N)

用它计算序列的 DTFT 时,输入变元 b 就是序列 x，其位置向量限定从零开始，即 n=0:(length(b)-1)。输入变元 a 取 1，变元 N 是把 $0\leqslant\omega<\pi$ 分割的份数，相当于本节中的 K/2。因为它计算的是正频率部分的特性，所以输出的频谱函数 H 和对应的频率向量 w 都只是正半频段的。若省略 N，则程序中默认的 N 值为 512。如果要求出全部正负频率的响应，那就应当键入第四个变元'whole'：[H,w] = freqz (b,a,N,'whole')。freqz 函数既可以用来计算序列的 DTFT，也可用来计算系统的频率特性。计算频率特性时的用法将在第 3.3.2 节中介绍。

3.2.8 MATLAB 计算举例

例 3.2.5：求有限长序列 $x(n)$=[1, 3, 5, 3, 1]的 DTFT，画出它在 ω=-8～8 rad/s 范围内的频率特性，讨论其对称性。再把 $x(n)$的位置向量的零点左右移动，讨论时移对 DTFT 的影响。

解：由于 x 序列的第二项下有双线，表示它的位置向量为 nx=[-1, 0, 1, 2, 3]，按定义

$$X(\mathrm{j}\omega)=\sum_{n=-\infty}^{\infty}x[n]\mathrm{e}^{-\mathrm{j}\omega n}=\mathrm{e}^{\mathrm{j}\omega}+3+5\mathrm{e}^{-\mathrm{j}\omega}+3\mathrm{e}^{-\mathrm{j}2\omega}+\mathrm{e}^{-\mathrm{j}3\omega}\quad -8\leqslant\omega\leqslant 8$$

现用 MATLAB 对它进行计算，将 ω在-8～8 rad/s 之间分为 1 000 份，列出程序 hc325 如下：

```
x=[1,3,5,3,1];nx=[-1:3];
w=linspace(-8,8,1000);                          % 设定频率向量
X=x*exp(-j*nx'*w);                              % 用 (3.2.1 a) 式计算 DTFT
subplot(5,3,1),stem(nx,x),axis([-2,6, -1,6])    % 画序列图
subplot(5,3,4),plot(w,abs(X)), subplot(5,3,7),plot(w,angle(X))    % 画幅频相频曲线
subplot(5,3,10),plot(w,real(X)), subplot(5,3,13),plot(w,imag(X))  % 画实频虚频曲线
nx1=nx+2; X1=x*exp(-j*nx1'*w);                  % 使 x 右移两位后，计算其 DTFT
nx2=nx-1; X2=x*exp(-j*nx2'*w);                  % 使 x 左移一位后，计算其 DTFT
% 对 nx1; X1 及 nx2; X2 采用相似的绘图语句
```

程序运行的结果见图 3.2.4，从中可以验证许多重要概念：

（1）序列的 DTFT 是连续函数。时域的离散序列应该对应于频域的连续频谱。

（2）序列的 DTFT 是周期函数，周期为 2π。通常只需知道它在$[-\pi,\pi]$,即奈奎斯特频率范围内的值，就知道了它在全部频率范围内的值。本题中为了观察其周期性，有意把频率区间取得较宽，以后数字频率 ω 将只取奈奎斯特频率$[-\pi,\pi]$范围，对应的模拟频率的范围为$[-\pi,\pi]F_s$或$[-\pi,\pi]/T$。

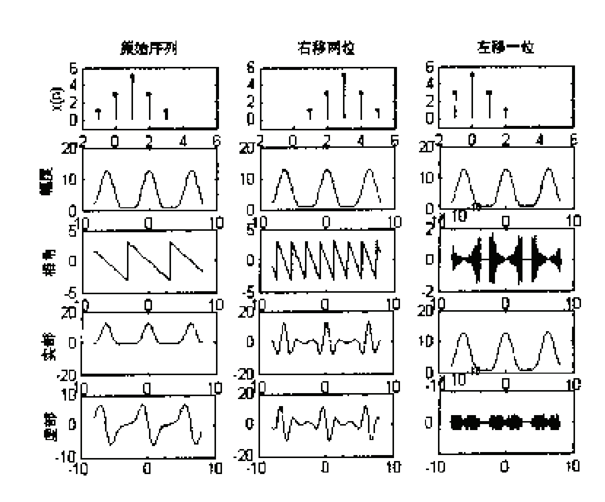


图 3.2.4　对称有限序列的频谱特性及时移影响

（3）由于本题给出的是实序列，实序列的 DTFT 具有对称性，其幅频特性和实频特性是偶对称的，而相频特性和虚频特性是奇对称的，因此在本书中，在只涉及幅频特性的问题上（例如滤波器设计），研究的范围往往只需考虑正

频率范围[0,π]或 [0,π]/T，但实际的频谱包括正负两边。负频率区的幅特性与正频率区对称，而其相频特性则是反对称的。

（4）信号在时间轴上的平移不影响其 DTFT 的幅频特性，只影响它的相频特性。这从时移公式中就可以看出。它只是把原来的 DTFT 乘以 $e^{-j\omega n_0}$，即右移 n_0 位，产生一个与频率成线性关系的负相移 $-\omega n_0$。

（5）本题给出的是时域对称的序列，它具有相位随频率线性变化的特点。对称中心所处的位置决定了相频特性的斜率的大小。当对称中心位于 n=0 处时（左移一位情况），相角恒为零，当然其斜率也为零（注意右 3 图和 5 图的纵坐标为 10^{-16}，图上标不清楚）；按对于题给的序列和右移两位的情况，对称中心分别位于 n=1 和 3 处，其斜率分别为 -1 和-3，从图上可以量出。左 3 图频率从-π 变到 π 时，相角从 π 变到-π；中 3 图频率从-π 变到 π 时，相角从 π 变到-π 共变了三个循环。

例 3.2.6：本例将用有限长实序列来验证线性特性(3.2.4)。令 $x_1(n)$ 和 $x_2(n)$为两个正态分布的随机序列，它们定义在 $0\leqslant n\leqslant 10$ 区间内。设它们的线性组合为

$$x(n)=\alpha\bullet x_1(n)+\beta\bullet x_2(n)$$

可以用数值方法分别计算它们的离散时间傅里叶变换 $X(\omega)$，$X_1(\omega)$和 $X_2(\omega)$。如果证明：X=alpha*X1+beta*X2 或 X-(alpha*X1+beta*X2)=0，就说明 DTFT 具有线性线性性质。

解：按上述思路，编出程序 hc326 如下：

```
x1 = randn(1,11); x2 = rand(1,11); n = 0:10;
alpha = 2; beta = 3;
k = 0:500; w = (pi/500)*k;
X1 = x1 * (exp(-j*pi/500)).^(n'*k);          % x1 的 DTFT
X2 = x2 * (exp(-j*pi/500)).^(n'*k);          % x2 的 DTFT
x = alpha*x1 + beta*x2;                      % x1 和 x2 的线性组合
X = x * (exp(-j*pi/500)).^(n'*k);            % x 的 DTFT
% 校验
Xc = alpha*X1 + beta*X2;                     % 把X1 和 X2 的线性组合称为 Xc
error = max(abs(X-Xc))                       % 全频段内最大误差绝对值
```

程序运行结果为：error = 7.1607e-015。

因为最大相对误差不到 10^{-14}，在 MATLAB 的计算精度下，这两个数组是一样的。

例 3.2.7：取一个复数序列 $x(n)=(0.8)^n e^{jn\pi/3}$，$0\leqslant n\leqslant 8$。求它的 $X(j\omega)$并探讨其对称性与周期性。

解：$x(n)$是复值的,又因 e 的指数项中只含虚数，它满足周期性，周期为 2π。计算时，可把范围取宽一些，到[-2π,2π]之间，以便观察其周期性。在这个区间中取 401 个频点作计算。另外，试着去掉 $x(n)$中的复数项，看它的 DTFT 与原来的 DTFT 有何关系？按此思路，编成如下程序 hc327。

```
n = 0:10; x = (0.8*exp(j*pi/3)).^n;
k = -200:200; w = (pi/100)*k;
X = x * (exp(-j*pi/100)) .^ (n'*k);           % 计算 x 的 DTFT
x1 = (0.8).^n;                                % 去掉 x 中的复指数部分，得到 x1
X1 = x1 * (exp(-j*pi/100)) .^ (n'*k);         % 计算 x1 的 DTFT
subplot(2,2,1); plot(w/pi,abs(X));grid        % 频率坐标用 π 为单位
subplot(2,2,3); plot(w/pi,angle(X));grid
subplot(2,2,2); plot(w/pi,abs(X1));grid
subplot(2,2,4); plot(w/pi,angle(X1));grid
```

运行程序 hc327 的结果见图 3.2.5。从所得曲线可以看到：

（1）复序列的 DTFT 的幅特性和相特性对 ω 都是周期性的，但都不是对称的，既不是偶对称，也不是奇对称。

（2）$e^{jn\pi/3}$ 是这个复序列的复数因子，去掉此因子，x 就成为实序列。实序列的幅特性和相特性对 ω 都是周期性的。其幅特性偶对称，相特性奇对称，因而 DTFT 是共轭对称的。

（3）把复序列变为实序列，相当于让时间序列乘以 $e^{-jn\pi/3}$。它引起 DTFT 特性的频移。这个实例也验证了这个性质，频移量为 $\pi/3$，这在图上也可得到验证。

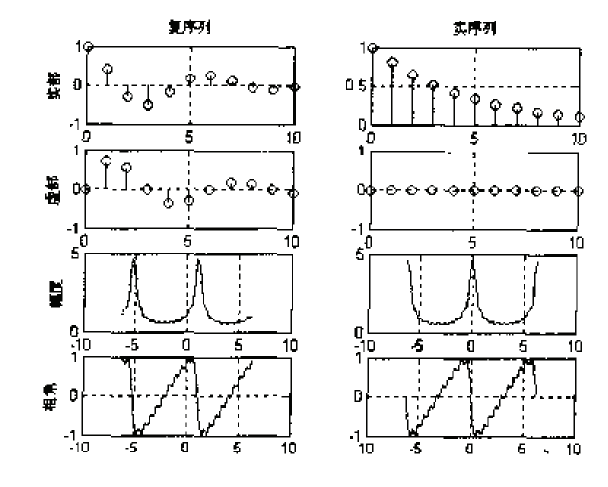


图 3.2.5　复序列（左）和实序列（右）频率特性的对称性

例 3.2.8：给定理想低通滤波器的频率响应为

$$H_d(\omega)=\begin{cases}1 & |\omega|\leqslant\omega_c\\ 0 & \omega_c\leqslant\omega\leqslant\pi\end{cases} \tag{3.2.31}$$

设 $\omega_c=1$，要求用 IDTFT 求出它的反变换对应的离散时间序列。假设采样周期为 T=1。

解：利用 IDTFT 公式（3.2.2 a）

$$\begin{aligned}h_d[n]=F^{-1}[X(j\omega)]&=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(j\omega)e^{j\omega n}d\omega=\frac{1}{2\pi}\int_{-\omega_c}^{\omega_c}e^{j\omega n}d\omega\\&=\frac{e^{j\omega_c n}-e^{-j\omega_c n}}{2j\pi n}=\frac{\sin\omega_c n}{n\pi},\qquad n=0,\pm1,\cdots\end{aligned} \tag{3.2.32}$$

将 $H_d(\Omega)$频谱图及 h 随 n 变化的曲线画成图 3.2.6。虽然在计算时采用了解析的方法，得到的是解的闭合形式，在画图时仍可利用 MATLAB。在 n=0 处出现的不定式可以用偏移微量的方法来处理，请读者自行完成。

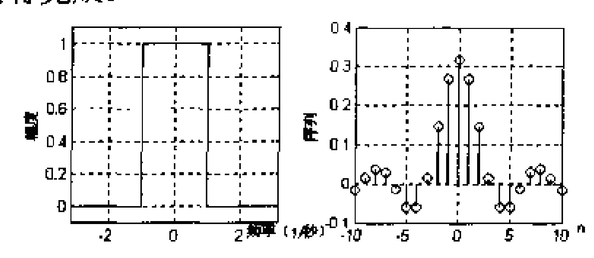


图 3.2.6　理想低通滤波器的频率特性（左）及其脉冲响应（右）

3.3　离散系统的频率特性

前面讨论的是离散信号的频谱特性，本节将讨论离散系统的频率特性，即在频域研究离散系统对离散序列的作用，研究离散信号通过离散系统后，应该如何在频率域计算其输出。

因为离散时间信号可以看成许多角频率不同的正余弦信号的合成。如果知道系统对不同频率的正余弦信号的响应，就可以用叠加原理来分析复杂信号所引起的输出。因为正余弦信号可以用复指数表示，下面将用复指数来讨论系统的频率特性。

3.3.1 定义

LTI 系统的一个主要特性是对于某些类型的输入特征函数，其输出仍然保持同样的特征函数，只是把输入函数乘了一个复数常数。正余弦函数和复指数函数就是这种特征函数之一。回顾第 2 章中(2.5.3)式所描述的系统的输入输出卷积关系如（3.3.1）式所示。

$$y(n)=\sum_{m=-\infty}^{\infty}h(m)x(n-m) \tag{3.3.1}$$

其中：$y(n)$和 $x(n)$分别是系统的输出和输入序列；$h(n)$是系统的脉冲响应。

如果输入是复指数序列：

$$x(n)=\mathrm{e}^{\mathrm{j}\omega n} \qquad -\infty<n<\infty \tag{3.3.2}$$

代入（3.3.1）式，得到

$$y(n)=\sum_{m=-\infty}^{\infty}h(m)x(n-m)=\left(\sum_{m=-\infty}^{\infty}h(m)\mathrm{e}^{-\mathrm{j}\omega m}\right)\mathrm{e}^{-\mathrm{j}\omega n} \tag{3.3.3}$$

写成
$$y(n)=H(\mathrm{e}^{\mathrm{j}\omega})x(n) \tag{3.3.4}$$

其中
$$H(\mathrm{e}^{\mathrm{j}\omega})=\sum_{m=-\infty}^{\infty}h(m)\mathrm{e}^{-\mathrm{j}\omega m} \tag{3.3.5}$$

从(3.3.4)式可以看出，复指数输入序列$\mathrm{e}^{\mathrm{j}\omega n}$所产生的输出序列 $y(n)$是具有同样频率的复指数函数，只是乘了一个复数常数$H(\mathrm{c}^{\mathrm{j}\omega})$。所以$\mathrm{e}^{\mathrm{j}\omega n}$就是离散 LTI 系统的特征函数之一。

上面定义的$H(\mathrm{e}^{\mathrm{j}\omega})$称为离散 LTI 系统的频率响应，它描述了系统在频域的行为。而(3.3.5)式表明，它就是系统的脉冲响应 $h(n)$的离散傅里叶变换(DTFT)。$H(\mathrm{e}^{\mathrm{j}\omega})$是$\omega$的复函数，其周期为 2π。它可以分解为实部和虚部，也可分解为幅度和相角，如（3.3.6）式所示。

$$H(\mathrm{e}^{\mathrm{j}\omega})=H_{\mathrm{re}}(\mathrm{e}^{\mathrm{j}\omega})+\mathrm{j}\cdot H_{\mathrm{im}}(\mathrm{e}^{\mathrm{j}\omega})=\left|H(\mathrm{e}^{\mathrm{j}\omega})\right|\mathrm{e}^{\mathrm{j}\theta(\omega)} \tag{3.3.6}$$

其中$H_{\mathrm{re}}(\mathrm{e}^{\mathrm{j}\omega})$和$H_{\mathrm{im}}(\mathrm{e}^{\mathrm{j}\omega})$分别是$H(\mathrm{e}^{\mathrm{j}\omega})$的实部和虚部，$\left|H(\mathrm{e}^{\mathrm{j}\omega})\right|$是离散系统的幅度响应而$\theta(\omega)$是它的相角响应。在有些情况下，幅度特性用分贝（dB）表示：

$$L=20\log_{10}\left|H(\mathrm{e}^{\mathrm{j}\omega})\right| \tag{3.3.7}$$

注意：增益和相位是 ω 的实函数，而频率响应则是 ω 的复函数。当输入序列是实数序列时，增益是 ω 的对称函数，而相位是 ω 的反对称函数（见表 3.6.1）。

上面是指数单频信号输入的情况。如果输入信号 $x(n)$是满足 DTFT 变换条件的任意信号，则可对（3.3.1）式两端做 DTFT，得到：

$$\begin{aligned}Y(\mathrm{e}^{\mathrm{j}\omega})&=\sum_{n=-\infty}^{\infty}y(n)\mathrm{e}^{-\mathrm{j}\omega n}=\sum_{n=-\infty}^{\infty}\left(\sum_{m=-\infty}^{\infty}h(m)x(n-m)\right)\mathrm{e}^{-\mathrm{j}\omega n}\\&=\sum_{m=-\infty}^{\infty}h(m)\left(\sum_{n=-\infty}^{\infty}x(n-m)\mathrm{e}^{-\mathrm{j}\omega(n-m)}\right)\mathrm{e}^{-\mathrm{j}\omega m}\\&=\sum_{m=-\infty}^{\infty}h(m)\left(\sum_{l=-\infty}^{\infty}x(l)\mathrm{e}^{-\mathrm{j}\omega l}\right)\mathrm{e}^{-\mathrm{j}\omega m}\end{aligned}$$

等式右端括号内就是 $x(n)$的 DTFT，式中两个下标变量 l 和 m 也可以完全分离，由此得到

$$Y(e^{j\omega}) = \left(\sum_{m=-\infty}^{\infty} h(m)e^{-j\omega m}\right) X(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega}) \tag{3.3.8}$$

其中 $H(e^{j\omega})$ 是（3.3.5）式定义的离散 LTI 系统的频率响应。（3.3.8）式给出了 LTI 系统在频域的输入输出关系，并且也给出了时域卷积对应于频域相乘的一个证明。由（3.3.8）式可以得到

$$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})} \tag{3.3.9}$$

因此离散时间LTI系统的频率响应 $H(e^{j\omega})$ 也可以由输出序列 $y(n)$ 的DTFT与输入序列 $x(n)$ 的 DTFT 之比求得。在第 2 章中讲到，离散时间 LTI 集总参数系统可以用差分方程描述如下：

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{i=0}^{M} b_i x(n-i) \qquad a_0 = 1 \tag{3.3.10}$$

对两边进行DTFT，利用DTFT的移位特性(3.2.5)式，$F[x(n-k)] = e^{-j\omega k}F[x(n)] = e^{-j\omega k}X(e^{j\omega})$，(3.3.10)式成为

$$Y(e^{j\omega})\sum_{k=0}^{N} a_k e^{-j\omega k} = X(e^{j\omega})\sum_{i=0}^{M} b_i e^{-j\omega i} \qquad a_0 = 1$$

因此

$$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})} = \frac{b_0 + b_1 e^{-j\omega} + b_2 e^{-j2\omega} + \cdots + b_M e^{-jM\omega}}{1 + a_1 e^{-j\omega} + a_2 e^{-j2\omega} + \cdots + a_N e^{-jN\omega}} \tag{3.3.11}$$

可见离散时间LTI集总参数系统的 $H(e^{j\omega})$ 可以用两个 $e^{-j\omega}$ 的多项式之比来表述，将不同的 ω 代入，求出这两个多项式的值，将它们相除，就可以求出它的频率特性。

3.3.2 用 MATLAB 计算系统频率特性

利用前面例子中计算离散时间傅里叶变换的方法，就可以用 MATLAB 分别求出 $H(e^{j\omega})$ 中分子分母多项式的 DTFT，它们是以频率为自变量的复数数组，其长度取决于所选频率的点数。然后将分子分母的两个复数数组“点除”，得出系统的频率特性。

由于频率特性的计算用得特别频繁，所以需要专门的计算函数。3.2.6 节中介绍的计算序列 DTFT 的函数 freqz 也是专门用来求离散时间系统频率特性的，所以结尾用字母 z；对连续系统的有类似的求频率特性函数 freqs，其结尾是字母 s，不要弄错。

调用 freqz 函数的方式如下。

[H,w]=freqz(b,a,N) 或 N 缺省[H,w]=freqz(b,a)。

输出变元 H 是系统的频率特性，它是一个 N 元的复数向量；

输出变元 w 是数字频率向量，它把 0 到 π 均分为 N 份，分辨率 π/N，w=[0:N-1]π/N；

输入变元 b 和 a 分别为分子分母多项式的负幂系数向量，即多项式的首项应是常数项，以后按 $e^{-j\omega}$ 的升幂排列，由此形成的多项式的系数向量；

输入变元 N 为所选的频率点数，它决定了频率分辨率的密度。N 的默认值为 512。

这样求出的频率特性是在正频率区间的特性，如果要得到全奈奎斯特频率范围的特性，就要运用 DTFT 共轭对称性质，使负频率段的幅特性与正频率段对称取值，而使其相特性与正频率段反对称取值。如果要让 MATLAB 计算全频段的频率特性，要增加一个输入变元'whole'：[H,w]=freqz(b,a,N,'whole')。

这时 N 个频点将在 0 到 2π 之间均分(不含 2π)，而得出的也将是奈奎斯特全频率范围的特性。

若没有左端变量，键入 freqz(b,a,N)，MATLAB 将不给出数据而只绘制频率特性。

若输出变元中只有 H，而在输入变元中给出频点向量 W：H=freqz(b,a,W)，这种调用方法可以在自己选定的频点向量 W 上计算频率特性，不采用全频段均分的频点。

采用模拟频率进行调用：键入[H,f]=freqz(b,a,N,Fs)，这种调用方法不采用数字频率 w，而采用模拟频率 f（单位为 Hz）。所以输入变元要有采样频率 Fs。

上面介绍的是最常用的调用方法，要进一步了解，可用 help freqz 获得。

例 3.3.1：设一个 LTI 系统的差分方程为 $y(n)-0.9y(n-1)=0.5x(n)+0.8x(n-1)$。

求出它的频率响应 $H(\mathrm{j}\,\omega)$，画图。并求输入为 $x(n)=\cos(0.1\pi n)u(n)$时的稳态输出 y_s。

解：按(3.3.11)式，可直接写出 $H(\mathrm{e}^{\mathrm{j}\omega})=\dfrac{Y(\mathrm{e}^{\mathrm{j}\omega})}{X(\mathrm{e}^{\mathrm{j}\omega})}=\dfrac{0.5+0.8\mathrm{e}^{-\mathrm{j}\omega}}{1-0.9\mathrm{e}^{-\mathrm{j}\omega}}=0.5\dfrac{1+1.6\mathrm{e}^{-\mathrm{j}\omega}}{1-0.9\mathrm{e}^{-\mathrm{j}\omega}}$

用下面的简单程序 hc331 绘出的 $H(\mathrm{j}\,\omega)$幅频与相频特性曲线如图 3.3.1 所示。

```
b=[0.5 0.8];  % 分子多项式系数向量
a=[1 -0.9];   % 分母多项式系数向量
[H,w]=freqz(B,A);  % 求出频率响应 (0 到 pi 分成 500 点)
subplot(2,1,1),plot(w,abs(H)),grid on
subplot(2,1,2),plot(w,angle(H)),grid on
```

稳态输入的频率为 $\omega_0=0.1\pi$，初始相角为 $\theta_0=0$，系统在该频点处的响应可由程序 hc331 的后半截来求：

```
w0=0.1*pi;
nw0=floor(w0/pi*512)          % 求出频率 w0 对应的下标 nw0
Hw0=H(nw0);                   % 求出频率 w0 处的频率响应
Aw0=abs(Hw0),                 % 求出频率 w0 处的幅特性
thetaw0=angle(Hw0)            % 求出频率 w0 处的相移
```

程序运行的结果为

nw0=51，Aw0 = 4.1919，thetaw0 = −1.2784

即 $H(\mathrm{e}^{\mathrm{j}0.1\pi})=\dfrac{0.5+0.8\mathrm{e}^{-\mathrm{j}0.1\pi}}{1-0.9\mathrm{e}^{-\mathrm{j}0.1\pi}}=4.1919\mathrm{e}^{-\mathrm{j}1.2784}$。所以：

$$y_s=4.2694\cos(0.1\pi\,n-1.2784)=4.1919\cos[0.1\pi\,(n-4.0693)]$$

下一段是绘制输入输出序列波形的语句。

```
n=0:30;x=cos(0.1*pi*n);
y=Aw0*cos(0.1*pi*n-thetaw0);
subplot(2,1,1),stem(n,x,'.')
subplot(2,1,2),stem(n,y,'.')
```

画出的波形见图 3.3.2。从图中可以看出输出的幅度比输入大了四倍多，而比较两者与横坐标交汇的零点，可以知道 y 比 x 延迟了约四个样本，在
包络周期中相当于 -1.2784rad 的负相移。

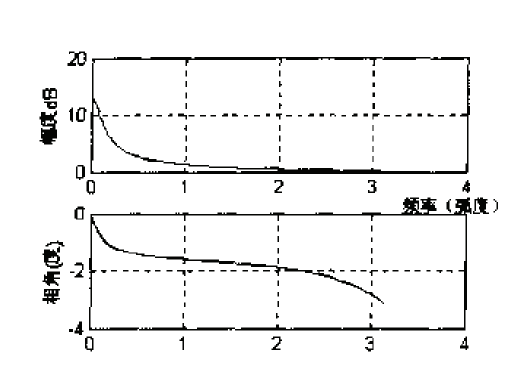


图 3.3.1 例 3.3.1 中系统的幅频和相频特性曲线

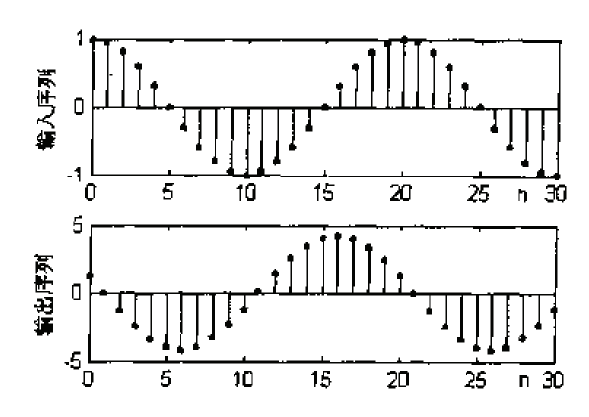


图 3.3.2 例 3.3.1 中的输入输出信号（放大并延迟）

3.4 周期序列的频谱——离散傅里叶级数

周期序列不满足绝对可加条件，因此它的 DTFT 无界，或无定义。要分析其 DTFT，可以先取 K 个周期，以后可让 K 无限增大而研究其极限，我们用算例说明此思路。

3.4.1 从一个计算实例出发

把例 3.2.2 中的四点序列加以周期延拓，画出多周期下的序列频谱。这可以用下列 MATLAB 程序方便地实现。其中延拓的周期数可以由读者自选。程序 fgp341 如下：

```
x0=[2,-1,1,1]; Nx=length(x0);                          % x0 是 4 点行向量
Nw=1000; dw=2*pi/Nw;    % 把 2π 分为 Nw 份，求频率分辨率 dw
k=floor((-Nw/2+0.5):(Nw/2-0.5));                       % k*dw 是正负对称的 Nw 点频率向量
for r=1:4
   K=input('延拓周期数 K=（建议依次取 1,10,100,500 四种）')
nx=0:(K*Nx-1);                                         % 延拓后的位置向量 nx
x=x0(mod(nx,Nx)+1);                                    % 延拓后的时域信号 x
   X=x*exp(j*dw*nx'*k);                                % 用公式(3.2.30)求 x 的 DTFT
   subplot(4,1,r),plot(k*dw,abs(X)),grid,shg           % 画图
   ylabel('abs(X(\omega))'),xlabel('\omega')
end
```

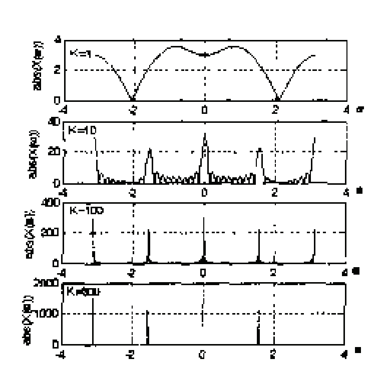


图 3.4.1 四点序列周期延拓后的频谱（依次为 1 10 100 500 个周期）

从图 3.4.1 可以看出：随着周期重复的次数增加，其频谱逐渐向几个频点集中。频点的数目为 N，其位置平均分布于奈奎斯特频率$[-\pi,\pi]$的范围内，相互间隔为 $2\pi/N$。频谱的幅度

也随周期重复的次数呈正比增大，最后趋向于幅值无穷的脉冲函数。在本例中在 w=0 处是信号中的直流分量，因为四点序列的代数和为 2-1+1+1=3，有一个较大的直流成分。另一个是正余弦分量，它同时具有正频率和负频率部分，此例中它位于 $\pi/2$ 处。这是可以想像到的，因为它的一个周期中只有四个样本点，所以它无法反映更高的频率成分。

如果把频谱的幅度除以延拓的周期数，在延拓的周期数趋向无穷大时，序列趋向于周期序列，这个幅度将趋向一个复数常数。从本例的数据中，可以发现，其频谱函数除以周期数应趋向于主值序列的 DTFT 在相应频率上的值。可以想像，它与傅里叶级数有关。

在程序中，建议的延拓周期数为 500，再增加这个数值会有问题。求 DTFT 的程序(5gp341)虽然非常简洁，但它的执行效率是很低的，在运行这个例题中延拓周期数=1000 时就会明显地感受到。在作者的目前的计算机上，大概要运行两分钟。如果输入延拓周期数=10 000，则计算机干脆回答"内存不足"而拒绝执行。

仔细分析一下就清楚了：以延拓 1000 个周期为例，x 的长度 nx 为 1000×4=4000，k 的长度已给定为 1000，矩阵 n'*k，因而矩阵 exp(j*dw*n'*k)将具有 1000×4000=4M 个复数元素。对于双精度数据，每个复数元素需要 16B 来存储，所以只存储这个矩阵就要 64MB 内存。而完成向量 x 与此矩阵的相乘法，也需要 1000×4000=4M 次复数乘法。周期数增加到 10 000 时，执行小小的（3.2.30）式需要 6.4 亿字节的内存和 4000 万次复数乘法。无论在存储空间和运行时间上都占用了过多资源，超越了大多数计算机和人的承受极限。至于实时应用，更不可想像。快速傅里叶变换就是为解决这个问题而提出来的。如果把程序中语句 X=x*exp(j*dw*nx'*k)换成 X=freqz(x,1,k*w)，由于后者采用了快速傅里叶变换，运算速度会大幅度地提高。

3.4.2 周期序列的傅里叶级数

设 $\tilde{x}(n)$ 为周期序列，其周期为 N。周期序列不满足绝对可加条件，按例题的思路，先取 K 个周期来分析。因此，n 求和的范围为 KN。按定义可知：

$$\tilde{X}_K(\omega)=\sum_{n=\langle KN\rangle}\tilde{x}(n)\mathrm{e}^{\mathrm{j}\omega n}$$

从上节的数字实例中看到，它的 DTFT 最后集中在 N 个频点 $\omega_k=2\pi k/N$，k=0,1,…,N-1 处，k 也可以向±∞延伸，但产生的数字频率仍然在奈奎斯特频率范围内。于是可以写成

$$\tilde{X}_K(k)=\tilde{X}_K(\omega_k)=\sum_{n=\langle KN\rangle}\tilde{x}(n)\mathrm{e}^{\mathrm{j}2\pi kn/N}=\sum_{n=\langle KN\rangle}\tilde{x}(n)W_N^{kn} \tag{3.4.1}$$

其中，引进下面的通用符号：

$$W_N=\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}} \tag{3.4.2}$$

它在几何上是一个相角为 $-2\pi/N$ 的单位向量。比如 $N=8$，相角的分度就等于 $2\pi/8$，$W_8=0.707-0.707\mathrm{j}$ 就是相角为-45° 的单位向量。

把(3.4.1)式的求和范围分成 m 段，每段的长度为 N，段内的位置序数为 l，即有

$$n=l+mN,\qquad m=0,\pm1,\cdots$$

把求和的过程也分段进行，先按 l 相加，即把每一周期内的 N 个样本加起来；再按 m 相加，即把 K 个周期的结果相加。再考虑到 $\tilde{x}(n)$ 的周期性

$$\tilde{x}(l)=\tilde{x}(l+mN)=x(l) \tag{3.4.3}$$

$x(l)$ 是 $\tilde{x}(l)$ 在主值区间的序列，其长度为 N。这样，(3.4.1)式可写成

$$\tilde{X}_K(k)=\sum_{m=\langle K\rangle}\sum_{l=0}^{N-1}\tilde{x}(l+mN)W_N^{k(l+mN)}=\sum_{m=\langle K\rangle}\left[\sum_{l=0}^{N-1}x(l)W_N^{kl}\right]W_N^{mN}$$

其中 $-\infty<k<\infty$。注意到方括号中求和号的结果与 m 无关，它是一个 N 元的常数序列。再注意到 $W_N^{mN}=\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}mN}=\mathrm{e}^{-\mathrm{j}2\pi\cdot m}=1$（当$m$为整数时），(3.4.1)式就成为

$$\tilde{X}_K(k)=K\cdot\left[\sum_{n=0}^{N-1}x(n)W_N^{kn}\right]=K\cdot\tilde{X}(k)$$

可见当 K 无限增大时，此 DTFT 确实无界。但若取 K=1，得到的

$$\tilde{X}(k)=\sum_{n=0}^{N-1}x(n)W_N^{kn}=\sum_{n=0}^{N-1}x(n)\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}kn} \tag{3.4.4 a}$$

却是有定义的。该序列除以 N 恰好是该周期序列的傅里叶级数系数 c_k，即

$$c_k=\frac{1}{N}\sum_{n=0}^{N-1}\tilde{x}(n)W_N^{kn}=\frac{1}{N}\tilde{X}(k)\qquad -\infty<k<\infty \tag{3.4.4 b}$$

所以，要定义周期序列的频谱，可以用它的傅里叶级数系数 c_k，看来也可以用 Nc_k。它就是在下节将讨论的离散傅里叶变换。式(3.4.4 a)中，k 和 n 均取整数，当 k 或者 n 变化时，W_N^{kn} 是周期为 N 的周期函数，可表示成 $W_N^{-\mathrm{j}k(l+mN)}=W_N^{-\mathrm{j}kl}$，$m$ 取整数。因此，系数 c_k 也是满足公式 $c_k=c_{k+mN}$ 的周期序列。可见，$\tilde{X}(k)$ 也是一个以 N 为周期的周期序列，称为 $\tilde{x}(n)$ 的离散傅里叶级数，用 DFS（Discrete Fourier Series）表示 。如对(3.4.4 a)式两端乘以 W_N^{kl}，并对 k 在一个周期中求和，得到：

$$\sum_{k=0}^{N-1}\tilde{X}(k)W_N^{kl}=\sum_{k=0}^{N-1}\left[\sum_{n=0}^{N-1}\tilde{x}(n)W_N^{kn}\right]W_N^{-kl}=\sum_{n=0}^{N-1}\tilde{x}(n)\sum_{k=0}^{N-1}W_N^{k(n-l)}$$

因为

$$\sum_{k=0}^{N-1}W_N^{k(n-l)}=\begin{cases}N & n-l=0\text{或}mN\\ 0 & n\neq l\end{cases} \tag{3.4.5}$$

可推出：

$$\tilde{x}(n)=\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kn}=\sum_{k=0}^{N-1}c_k\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kn} \tag{3.4.6}$$

(3.4.4 a)式和(3.4.6)式称为一对 DFS。(3.4.6）式表明，可以将周期序列分解成 N 次谐波，第 k 个谐波频率为 $\omega_k=(2\pi/N)k,\ k=0,1,2,\cdots,N-1$，幅度为 $(1/N)\tilde{X}(k)$。可见，周期序列可以用 DFS 表示它的频谱分布规律。

例 3.4.1：设 $x(n)=R_4(n)$，将 $x(n)$ 以 $N=8$ 为周期进行周期延拓，得到如图 3.4.2(a)所示的周期序列 $\tilde{x}(n)$，周期为 8，求 $\tilde{x}(n)$ 的 DFS。

解：按照(3.4.7)式有

$$\tilde{X}(k)=\sum_{n=0}^{7}\tilde{x}(n)\mathrm{e}^{-\mathrm{j}\frac{2\pi}{8}kn}=\sum_{n=0}^{3}\mathrm{e}^{-\mathrm{j}\frac{\pi}{4}kn}=\frac{1-\mathrm{e}^{-\mathrm{j}\frac{\pi}{4}k\cdot 4}}{1-\mathrm{e}^{-\mathrm{j}\frac{\pi}{4}k}}=\frac{1-\mathrm{e}^{-\mathrm{j}\pi k}}{1-\mathrm{e}^{-\mathrm{j}\frac{\pi}{4}k}}=\frac{\mathrm{e}^{-\mathrm{j}\frac{\pi}{2}k}(\mathrm{e}^{\mathrm{j}\frac{\pi}{2}k}-\mathrm{e}^{-\mathrm{j}\frac{\pi}{2}k})}{\mathrm{e}^{-\mathrm{j}\frac{\pi}{8}k}(\mathrm{e}^{\mathrm{j}\frac{\pi}{8}k}-\mathrm{e}^{-\mathrm{j}\frac{\pi}{8}k})}=\mathrm{e}^{-\mathrm{j}\frac{3}{8}\pi k}\frac{\sin\frac{\pi}{2}k}{\sin\frac{\pi}{8}k}$$

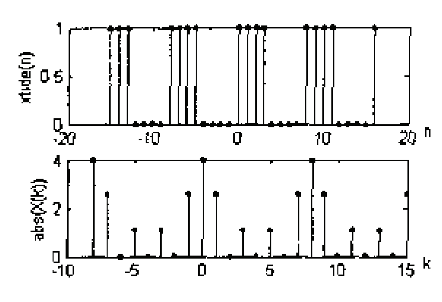


图 3.4.2　例 3.4.1 序列的 DFS

这个例题的傅里叶级数系数是可以解析地求得的，可以不用数值计算。不过照这个式子画图也有点麻烦：一是求指数，二是在求分子分母两个正弦同时过零点处要使自变量偏移。这里还是用 MATLAB 帮助绘图，程序如下。

```
k=[-8:15]+eps;
X=exp(-j*3/8*pi*k).*sin(pi/2*k)./sin(pi/8*k)
stem(k,abs(X),'.')
```

画出的幅度特性 $\left|\tilde{X}(k)\right|$ 如图 3.4.2 所示。

3.4.3　用 MATLAB 计算离散傅里叶级数

(3.4.4b)和 (3.4.6)式构成的一对离散傅里叶级数可以很简捷地用 MATLAB 程序计算。先用矩阵来表示(3.4.4b) 式，可以把它写成：

$$\tilde{X}=\begin{bmatrix}\tilde{X}(1)\\ \tilde{X}(2)\\ \vdots\\ \tilde{X}(N-1)\end{bmatrix}=\begin{bmatrix}1 & 1 & 1 & \cdots & 1\\ 1 & W_N^{\ 1} & W_N^{\ 2} & \cdots & W_N^{\ N-1}\\ 1 & W_N^{\ 2} & W_N^{\ 4} & \cdots & W_N^{\ 2(N-1)}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 1 & W_N^{\ N-1} & W_N^{\ 2(N-1)} & \cdots & W_N^{\ (N-1)(N-1)}\end{bmatrix}\begin{bmatrix}\tilde{x}(0)\\ \tilde{x}(1)\\ \vdots\\ \tilde{x}(N-1)\end{bmatrix}=W\cdot\tilde{x} \quad (3.4.7)$$

这个矩阵运算可用下列几条 MATLAB 语句来实现。可以把它写成求 DFS 的子程序。

```
n = [0:1:N-1]; k = [0:1:N-1];      % 设定n和k的行向量
WN = exp(-j*2*pi/N);               % 设定Wn因子
nk = n'*k;                         % 产生一个含nk值的N乘N维的整数矩阵
WNnk = WN .^ nk;                   % 求出W矩阵
Xk = xn * WNnk;                    % 求出离散傅里叶级数系数
```

前两条语句无须解释。第三条语句把 n 转为列向量 n'，再与行向量 k 做矩阵乘，它产生的是一个整数矩阵：

$$\text{n'*k}=\begin{bmatrix}0\\ 1\\ 2\\ \vdots\\ N-1\end{bmatrix}[0,1,2,\cdots,N-1]=\begin{bmatrix}0 & 0 & 0 & \cdots & 0\\ 0 & 1 & 2 & \cdots & N-1\\ 0 & 2 & 4 & \cdots & 2\times(N-1)\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & N-1 & (N-1)\times 2 & \cdots & (N-1)\times(N-1)\end{bmatrix} \quad (3.4.8)$$

这个整数方阵恰好是算式 $\tilde{X}=W\cdot\tilde{x}$ 中 W 矩阵的指数部分。第四条语句就产生了 W 矩阵，而最后一条语句就完成了 DFS 的全部运算。由于实际上离散傅里叶级数并不太用到。它已被下节将介绍的离散傅里叶变换取代，而且两者的计算程序是完全相同的。所以如果写成 MATLAB 子程序，它可以命名为 DFS，也可以命名为 DFT。

反变换 IDFS 或 IDFT 的 MATLAB 程序与正变换相仿，它的矩阵形式为：

$$\tilde{x}=\begin{bmatrix}1 & 1 & 1 & \cdots & \cdots & 1\\ 1 & W_N^{-1} & W_N^{-2} & \cdots & \cdots & W_N^{-(N-1)}\\ 1 & W_N^{-2} & W_N^{-4} & \cdots & \cdots & W_N^{-2(N-1)}\\ \vdots & \vdots & \vdots & \ddots & & \vdots\\ 1 & W_N^{-(N-1)} & W_N^{-2(N-1)} & & \cdots & W_N^{-(N-1)(N-1)}\end{bmatrix}\begin{bmatrix}\tilde{X}(0)\\ \tilde{X}(1)\\ \vdots\\ \tilde{X}(N-1)\end{bmatrix}=W^{-1}\cdot\tilde{X} \quad (3.4.9)$$

所以计算的 MATLAB 程序中只需把输入改成 Xk，输出改成 xn，保留程序的前三行，并将程序的最后两句改为：

```
IWNnk = WN .^ (-nk);          % 求出反W矩阵
xn = (Xk * IWNnk)/N;          % 求出xn
```

即可。由于 MATLAB 中有编好的 FFT 和 IFFT 子程序可以调用，所以不必自编反变换子程序了。

3.5 离散傅里叶变换(DFT)

离散傅里叶级数对偶[见式(3.4.4 b)和(3.4.6)]提供了一种计算有限频点上频谱的技巧，但如果按公式(3.4.6)对离散傅里叶级数进行反变换，得到的是一个周期信号 $\tilde{x}(n)$。在实际中大多数信号具有有限持续时间，并不具有周期性。对这些信号，怎样探讨一种可数值计算的傅里叶表示式？人们就想到，根据有限长序列的基本形状，延拓为一周期信号，然后采用此周期信号的 DFS，只取其主值作为其频谱。这样就产生了一个新的变换，叫做离散傅里叶变换(DFT)，它是 DFS 的主周期。用这样的傅里叶变换就可对任意有限长序列进行数值计算。

3.5.1 离散傅里叶变换(DFT)的定义

有限长序列 $x(n)(0\leqslant n\leqslant N-1)$ 有 N 个样本值。它的傅里叶变换 $X(\mathrm{j}\omega)$ 在频率区间 $(0\leqslant\omega<2\pi)$ 的 N 个等间隔分布的点 $\omega k=2\pi k/N(0\leqslant k\leqslant N-1)$ 上也有 N 个取样值。这两组有限长的序列之间可以用简单的关系联系起来：

$$X(k)=X(\mathrm{e}^{\mathrm{j}\omega})\Big|_{\omega=2\pi k/N}=\sum_{n=0}^{N-1}x(n)\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}kn} \qquad 0\leqslant k\leqslant N-1 \qquad (3.5.1)$$

$X(k)$就称为 $x(n)$的离散傅里叶变换(DFT)。利用 3.5 节中(3.4.2)式的通用符号 $W_N=\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}}$，可把(3.5.1)式写成：

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{kn} \qquad 0\leqslant k\leqslant N-1 \qquad (3.5.2)$$

它的反变换称为离散傅里叶反变换(IDFT)，可以表为：

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-kn} \qquad 0\leqslant n\leqslant N-1 \qquad (3.5.3)$$

要证明上述反变换关系，可以把(3.5.3)式两端同乘以 W_N^{ln} 并将结果从 $n=0$ 加到 $n=N-1$，得到

$$\sum_{n=0}^{N-1}x(n)W_N^{nl}=\sum_{n=0}^{N-1}\frac{1}{N}\left[\sum_{k=0}^{N-1}X(k)W_N^{-kn}\right]W_N^{ln}=\frac{1}{N}\sum_{n=0}^{N-1}\sum_{k=0}^{N-1}X(k)W_N^{-(k-l)n}$$

改变等式右端求和的次序：

$$\sum_{n=0}^{N-1}x(n)W_N^{nl}=\frac{1}{N}\sum_{k=0}^{N-1}\sum_{n=0}^{N-1}X(k)W_N^{-(k-l)n}=\frac{1}{N}\sum_{k=0}^{N-1}X(k)\left[\sum_{n=0}^{N-1}W_N^{-(k-l)n}\right] \tag{3.5.4}$$

根据（3.4.5）式 $\sum_{n=0}^{N-1}W_N^{-(k-l)n}=\begin{cases}N & 对k-l=rN,r为整数\\0, & 其他\end{cases}$，上式可以化为

$$\sum_{n=0}^{N-1}x(n)W_N^{nl}=\sum_{k=0}^{N-1}X(k)\bigg|_{k=l}=X(l) \tag{3.5.5}$$

这就证明了(3.5.3)式与(3.5.2)式确实是一对对偶关系式。用矩阵来表示，可写成：

$$X=\begin{bmatrix}X(0)\\X(1)\\\vdots\\X(N-1)\end{bmatrix}=\begin{bmatrix}1 & 1 & 1 & \cdots & 1\\1 & W_N^{1} & W_N^{2} & \cdots & W_N^{N-1}\\1 & W_N^{2} & W_N^{4} & \cdots & W_N^{2(N-1)}\\\vdots & \vdots & \vdots & \ddots & \vdots\\1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)}\end{bmatrix}\begin{bmatrix}x(0)\\x(1)\\\vdots\\x(N-1)\end{bmatrix}=W\cdot x \tag{3.5.6}$$

(3.5.6)式与(3.4.7)式完全雷同，而求 IDFT 的算式则与(3.4.9)式相仿。它们可以用 3.4 节介绍的 MATLAB 程序来求解。

不难看出，计算每一个 DFT 分量 $X(i)$，需要做 N 次复数乘法和 N-1 次复数加法。而要完成整个 DFT，求出 X，则需要做 $N\times N$ 次乘法和 $N(N-1)$次复数加法。当 N 很大时,这个计算过程相当费时，另外，它占的数据存储量为 $N\times N$。在 MATLAB 中每个数据都是以双精度格式保存的，需要 8（字节），所以总共要 $8\times N\times N$ B。依照这个思路编出的程序，当然有同样的毛病。因此，当 N 较大时，MATLAB 会拒绝执行这个程序，并警告内存不足。为此，人们研究了很多种加快计算并节省内存的方法，统称为快速傅里叶变换(FFT)，在第 4 章中将对此进行讨论。

3.5.2 用 MATLAB 计算 DFT 和 IDFT

上面已经指出，按（3.4.7）式编成的简易的 DFT(或 DFS)程序只能用来说明概念，或作小规模的计算。在实用中，MATLAB 提供了内部函数来快速地进行离散傅里叶变换(DFT)计算。它采用了优化的算法，并且程序是用目的码编写的，使得它有极高的运行速度。因此它的命名就是 fft，而反变换就是 ifft。调用方法为 fft(x)或 fft(x,N)。

X=fft(x)完成对序列 x 的 L 点 DFT，其中 L 为序列 x 的长度。fft(x,N)则指定了采用 N 点 DFT。如果 $N>L$,则程序会自动给 x 后面补 N-L 个零；如果 $N<L$，则程序会自动将 x 截断，取前 N 个数据。同样，计算傅里叶反变换(IDFT)时，调用函数 ifft 方法为 ifft(x)或 ifft(x,N)。

其输入变元的意义与 fft 函数相同。

MATLAB 的信号处理工具箱还给出了函数 dftmtx(N)，它可以给出(3.4.7)式中求 DFT 的矩阵 W，求 IDFT 的矩阵 W^{-1} 可用 conj(dftmtx(N))/N=W^{-1} 求得。

例 3.5.1：求例 3.2.4 中 x=[2,-1,1,1]的离散傅里叶变换。并画到图上，与其 DTFT 比较。

解：在例 3.2.4 的程序中绘制幅特性的语句后加上 hold on，接着加上：

```
Xd=fft([2,-1,1,1]),plot([0:3]*2*pi/4,abs(Xd),'o')
```

运行后将得到图 3.5.1(a)的曲线。可以看到，左边两个 DFT 点与 DTFT 曲线重合，而右边两个 DFT 点跑出界了。因为用 FFT 求出的频谱横轴，取的是 $0\leqslant\omega<2*\pi$，而 DTFT 则取 $-\pi\leqslant\omega<\pi$。所以要把 w 轴上 $\pi\leqslant\omega<2*\pi$ 频段的频谱循环移位到 $-\pi\leqslant\omega<0$ 频段上去。

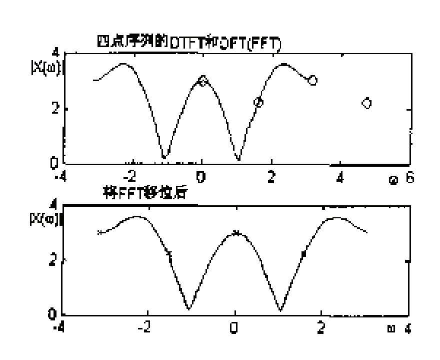


图 3.5.1　DTFT 与 DFT 及 FFT 的对应关系

MATLAB 设计了专门的函数 fftshift(Xd)可把 Xd 的后半段移到前面去，但同时还必须写一条语句把位置向量的后半段也移到前面去，即把上述 plot 语句中的[1:3]*2*pi/4 换成[-2:1]*2*pi/4[见(3.2.19)式]。在图 3.5.1 中的第二子图上重画 DTFT 曲线并画出移位后的 FFT 的语句如下：

```
Xd1=fftshift(Xd);
subplot(2,1,2),plot(k*dw,abs(X)),hold on      % 重画DTFT 曲线
plot([-2:1]*2*pi/4,abs(Xd1),'x')              % 画出移位后的FFT
```

运行后将得到图 3.5.1(b)的曲线。可见 DFT 实际上是 DTFT 连续频域曲线上的离散样本点。

3.5.3　两种离散傅里叶变换(DFT 与 DTFT)的关系

（1）通过插值由 DFT 求 DTFT

由(3.5.1)式可知，长度为 N 的序列 $x(n)$的 N 点 DFT $X(k)$,实际上就是序列的 DTFT 在 N 个等间隔频点 $\omega k=2\pi k/N\,(0\leqslant k\leqslant N-1)$ 上的样本 $X(k)$。假如系统确是 N 阶，则知道了 N 点的 $X(k)$，就可以惟一地确定序列 DTFT $X(\mathrm{e}^{\mathrm{j}\omega})$。方法是采用图 3.5.2 的流程。先用 IDFT 公式(3.5.3)由 $X(k)$求 $x(n)$,再由(3.2.1)式及 $x(n)$求 $X(\mathrm{e}^{\mathrm{j}\omega})$ 如下：

$$
\begin{aligned}
X(\mathrm{j}\omega)&=\sum_{n-\infty}^{\infty}x(n)\mathrm{e}^{-\mathrm{j}\omega n}=\frac{1}{N}\sum_{n-\infty}^{\infty}\left[\sum_{k=0}^{N-1}X(k)W_N^{-kn}\right]\mathrm{e}^{-\mathrm{j}\omega n}\\
&=\frac{1}{N}\sum_{k=0}^{N-1}X(k)\sum_{n=0}^{N-1}\mathrm{e}^{\mathrm{j}2\pi kn/N}\bullet\mathrm{e}^{-\mathrm{j}\omega n}
\end{aligned}
\tag{3.5.7}
$$

X(k) → IDFT → X(k) → DIFT → X(jω)

图 3.5.2　由 N 点离散 DFT 求连续频谱 DTFT 的流程

等式右端第二个求和号可以化简成正余弦关系：

$$
\sum_{n=0}^{N-1}\mathrm{e}^{-\mathrm{j}(\omega-2\pi k/N)n}=\frac{1-\mathrm{e}^{-\mathrm{j}(\omega N-2\pi k)}}{1-\mathrm{e}^{-\mathrm{j}[\omega-(2\pi k/N)]}}=\frac{\mathrm{e}^{-\mathrm{j}(\omega N-2\pi k)/2}}{\mathrm{e}^{-\mathrm{j}[(\omega N-2\pi k)/2N]}}\bullet\frac{\sin[(\omega N-2\pi k)/2]}{\sin[(\omega N-2\pi k)/2N]}
\tag{3.5.8}
$$

将它代入上式，最后得到：

$$
X(\mathrm{j}\omega)=\sum_{k=0}^{N-1}X(k)\bullet\frac{\sin[(\omega N-2\pi k)/2]}{N\sin[(\omega N-2\pi k)/2N]}\bullet\mathrm{e}^{-\mathrm{j}[\omega-(2\pi k/N)][(N-1)/2]}
\tag{3.5.9}
$$

（3.5.9）式实际上表示了一个内插关系，定义插值函数

$$\Phi(\omega)=\frac{\sin(N\omega/2)}{N\sin(\omega/2)}\cdot e^{-j\omega\left[\frac{N-1}{2}\right]} \tag{3.5.10}$$

则可以得出频率特性的内插公式：

$$X(j\omega)=\sum_{k=0}^{N-1}X(k)\cdot\Phi\left(\omega-\frac{2\pi k}{N}\right) \tag{3.5.11}$$

根据这个公式，可以由 N 个频点上的离散数据 $X(k),(k=1,\cdots,N)$,求得奈奎斯特频域内所有频点 ω 上的 $X(j\omega)$。这里的 ω 可以取更密的网格，以求得更连续的结果。

（2）通过增加序列长度使DFT趋近DTFT

在有MATLAB等计算软件的情况下，用插值运算就不见得是好办法。因为插值函数本身就很烦琐，而加大DFT的计算点数却很容易，在编程上只是改一个数。由于DFT的高效计算方法FFT在软件中是现成的，即使多达几万个点，所需的计算时间不过几秒。当原有时间序列很短时，有两种方法增加DFT的计算点数：一种是增加序列取值的长度，这可以增加输入信号的阶数，得到更加精密的频谱，如果可能，应尽量采取这种方法；如果输入序列取值的长度无法增加，那么只能在数据的后面补零来加大序列的长度。后一种方法实际上并没有增加数据的信息量，有效数据仍只有四个，序列仍是四阶的。在频谱上增加的点数只能增加实际的分辨率，并不能给出更详细的形状，举例如下。

例 3.5.2：将例3.5.1的情况再做一遍。还是输入那四个点，现希望得到具有64点的频谱，则程序如下：

```
xm=[2,-1,1,1,zeros(1,60)];
Nm=length(xm);nxm=0:Nm-1;
Xdm=fftshift(fft(xm));
kxm=floor(-Nm/2+0.5:Nm/2-0.5);          % 按(3.2.19)式确定频率位置向量
plot(kxm*2*pi/Nm,abs(Xdm),'o')
```

解：绘出的DFT见图3.5.3。可见它就是原来的DTFT上取出的64个样本点，只是fft的频率坐标的范围取了0～2*pi的一段，本程序通过fftshift函数把频谱序列的右半段平移到了左边，而通过语句kxm=floor(-Nm/2+0.5:Nm/2-0.5)把FFT的位置向量也移到图上的-pi～pi。如果把样本再加密，就可以把它看做连续曲线了。

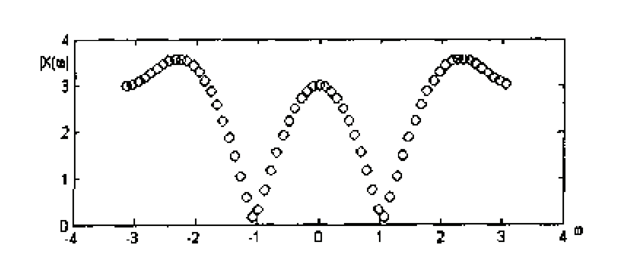


图3.5.3　例3.5.2的四点序列提高了视在分辨率的频谱

由此可见，DFT提供了对有限长度序列进行DTFT数值计算的实用方法。

3.6　离散傅里叶变换(DFT)的特性

在本章开始曾经指出，离散傅里叶变换(DFT)和离散时间傅里叶变换(DTFT)的主要区别就在于DFT在时域和频域都是离散的。它带来的最大好处就是适合于数值计算，适合于计算机处理。DFT和DTFT有许多相似的性质，所以与DTFT统一列于表3.6.1～3.6.3中（见本节末尾部）。

两者的主要不同点在于DFT定义在一个等间距划分成 N 个频点单位圆圆周上，到达 $N-1$

点以后，它就又回到零，因此频率和时间作为自变量都具有循环的特性。而 DTFT 的时间和频率轴都是可以无限延伸的。所以读者在阅读这本节时要把握住这一点，就容易掌握。DFT 的有些性质与 DTFT 性质的证明过程相似，此处就不再重复。

3.6.1　隐含周期性

前面定义的 DFT 变换对中，$x(n)$ 与 $X(K)$均为有限长序列，但由于W_N^{kn} 的周期性，使(3.5.1)式和(3.5.2)式中的 $X(K)$隐含了周期性，且周期均为 N 。对任意整数 m ，总有

$$W_N^k = W_N^{(k+mN)} \qquad k, m, N \text{ 均为整数}$$

所以(3.5.2)式中， $X(k)$ 满足 $X(k+mN)=\sum_{n=0}^{N-1}x(n)W_N^{(k+mN)n}=\sum_{n=0}^{N-1}x(n)W_N^{kn}=X(k)$

同理可证明(3.5.3)式中 $x(n+mN)=x(n)$ 。

实际上，任何周期为 N 的周期序列 $\tilde{x}(n)$ 都可以看做长度为 N 的有限长序列 $x(n)$ 的周期延拓，而 $x(n)$ 则是 $\tilde{x}(n)$ 的一个周期，即

$$\tilde{x}(n)=\sum_{m=-\infty}^{\infty}x(n+mN) \tag{3.6.1}$$

$$x(n)=\tilde{x}(n)\cdot R_N(n) \tag{3.6.2}$$

定义周期序列 $\tilde{x}(n)$ 中从 n=0 到 N-1 的第一个周期为 $\tilde{x}(n)$ 的主值区间，而主值区间上的序列称为 $\tilde{x}(n)$ 的主值序列。因此 $x(n)$ 与 $\tilde{x}(n)$ 的上述关系可叙述为：$\tilde{x}(n)$ 是 $x(n)$ 的周期延拓序列；$x(n)$ 是 $\tilde{x}(n)$ 的主值序列。

为了以后叙述方便，将(3.6.1)式用如下形式表示：

$$\tilde{x}(n)=x(n)_N \tag{3.6.3}$$

式中 $x(n)_N$ 表示 $x(n)$ 以 N 为周期的周期延拓序列，$(n)_N$ 表示 n 对 N 求余，即如果

$$n=mN+n_1 \quad 0\leqslant n_1\leqslant N-1，m \text{ 为整数}$$

则 $(n)_N=n_1$ 。与这个表达式等价的 MATLAB 语句为 `n1=mod(n,N)`。

如果 $x(n)$ 的长度为 N ，且 $\tilde{x}(n)=x(n)_N$ ，则可写出 $\tilde{x}(n)$ 的离散傅里叶级数表示式

$$\tilde{X}(k)=\sum_{n=0}^{N-1}\tilde{x}(n)W_N^{kn}=\sum_{n=0}^{N-1}x((n)_N)W_N^{kn}=\sum_{n=0}^{N-1}x(n)W_N^{kn} \tag{3.6.4}$$

$$\tilde{x}(n)=\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)W_N^{-kn}=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-kn} \tag{3.6.5}$$

式中 $X(k)=\tilde{X}(k)R_N(k)$ 为 $\tilde{X}(k)$ 的主值序列。将(3.6.4)式和(3.6.5)式与 DFT 定义(3.5.2)式和(3.5.3)式相比较可知，有限长序列 $x(n)$ 的离散傅里叶变换 $X(k)$ ，正好是 $x(n)$ 的周期延拓序列 $x((n)_N)$ 的离散傅里叶级数系数 $a_k\bullet N=\tilde{X}(k)$ 的主值序列。

3.6.2　线性性质

如果 $x_1(n)$ 和 $x_2(n)$ 是两个有限长序列，长度分别为 N_1 和 N_2 ，且 $y(n)=a\,x_1(n)+b\,x_2(n)$ ，式中 a 、b 为常数，取 $N=\max[N_1,N_2]$ ，则 $y(n)$ 的 N 点 DFT 为

$$Y(k)=\text{DFT}[\,y(n)\,]=a\,X_1(k)+b\,X_2[k] \qquad 0\leqslant k\leqslant N-1 \tag{3.6.6}$$

其中 $X_1(k)$ 和 $X_2(k)$ 分别为 $x_1(n)$ 和 $x_2(n)$ 的 N 点 DFT。

3.6.3　循环移位性质

（1）序列的循环移位

设 $x(n)$ 为有限长序列，长度为 N，则 $x(n)$ 的循环移 m 位的定义为

$$y(n) = x((n+m)_N)R_N(n) \tag{3.6.7}$$

(3.6.7)式表明，将 $x(n)$ 以 N 为周期进行周期延拓得到 $\tilde{x}(n) = x((n)_N)$，再将 $\tilde{x}(n)$ 左移 m 位得到 $\tilde{x}(n+m)$，最后取 $\tilde{x}(n+m)$ 的主值序列则得到有限长序列 $x(n)$ 的循环移位序列 $y(n)$。$x(n)$ 及其循环移位过程如图 3.6.1 所示。显然，$y(n)$ 仍是长度为 N 的有限长序列。观察图 3.6.1 可见，循环移位的实质是将 $x(n)$ 左移 m 位，而移出主值区 $0 \leqslant K \leqslant N-1$ 的序列值又依次从右侧进入主值区。“循环移位”就是由此而得名的。

例 3.6.1：设 x=[7,6,5,4,3,2]，位于主值区间。现要把 x 循环右移两位，成为新主值区间的向量 y，并画出循环移位的中间过程。

解：首先把 x 作为周期延拓，成为多周期向量 $x1$，设把它的位置向量左右各延长一个周期；然后将 $x1$ 右移两位，成为多周期向量 $y1$，再取出它的主值部分，得到 y，依次在同样横坐标下画出 x,$x1$,$y1$ 和 y 四个子图。则其 MATLAB 程序 hc361 如下：

```
x=[7,6,5,4,3,2];Nx=length(x);nx=0:Nx-1;        % x序列的参数
nx1=-Nx:2*Nx-1;x1=x(mod(nx1,Nx)+1);            % 延拓为周期向量x1,注意mod用法
[y1,ny1]=sigshift(x1,nx1,2);                   % 将x1右移两位, 得到y1
RN=(nx1>=0)&(nx1<Nx);                          % 在x1的位置向量nx1上设置主值窗
RN1=(ny1>=0)&(ny1<Nx);                         % 在y1的位置向量ny1上设置主值窗
subplot(4,1,1),stem(nx1,RN.*x1)                % 在子图上画出x1的主值
subplot(4,1,2),stem(nx1,x1)                    % 画出x1
subplot(4,1,3),stem(ny1,y1)                    % 画出y1
subplot(4,1,4),stem(ny1,RN1.*y1)               % 画出y1的主值
```

各子图都用 axis([-Nx,2*Nx-1,0,10])语句把坐标对齐后，得出的结果如图 3.6.1 所示。

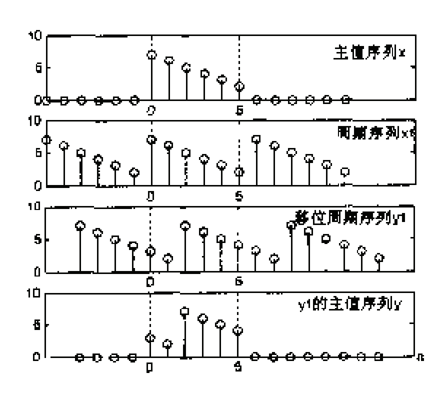


图 3.6.1　DFT 序列的循环移位过程

可以开发一个时域循环移位的 MATLAB 函数 cirshftt.m 如下:其中 y 为输出序列，x 为输入序列，其长度必须小于等于循环长度 N，m 为移位步数。实现 $y(n) = x((n-m) \bmod N)$。

```
function y = cirshftt(x,m,N)
if length(x) > N error('N 必须 >= x 的长度') end     % 检查x的长度应小于等于N
x = [x zeros(1,N-length(x))]; n = [0:1:N-1];         % 把x补到长度N
y = x(mod(n-m,N));                                   % 求出x的循环折叠序列y
```

例如，给定 x=[7,6,5,4,3,2]; N=6; m=2，调用这个子程序，得到 y=[3,2,7,6,5,4]。这和图 3.6.1 上最后一个子图完全一致。读者如果对“循环”的概念比较清楚，知道从右端移出的数据该进入左端。那么中间的两个子图完全没有必要画出，从程序中也可以不要与它们对应的语句。这就意味着直接对下标进行求余（mod）处理，这样的方法比较高明。后面将用这样的思路对循环折叠和循环卷积进行编程。

（2）时域循环移位定理

设 $x(n)$ 是长度为 N 的有限长序列，$y(n)$ 为 $x(n)$ 的 m 位循环移位序列，即 $y(n)=x((n+m)_N)R_N(n)$，则

$$Y(k)=DFT[\,y(n)]=W_N^{-km}X(k) \tag{3.6.8}$$

其中 $X(k)=DFT[\,x(n)]n,\quad 0\leqslant k\leqslant N-1$。

证明：$Y(k)=\mathrm{DFT}[y(n)]=\sum\limits_{n=0}^{N-1}x((n+m)_N)R_N(n)W_N^{kn}=\sum\limits_{n=0}^{N-1}x((n+m)_N)W_N^{kn}$，令 $n+m=n'$，则有

$$Y(k)=\sum_{n'=m}^{N-1+m}x((n')_N)W_N^{k(n'-m)}=W_N^{-km}\sum_{n'=m}^{N-1+m}x((n')_N)W_N^{kn'}=W_N^{-km}X(k)$$

由于上式中求和项 $x((n')_N)W_N{}^{kn'}$ 以 N 为周期，所以可以把求和周期移到主值区间而不影响最后结果，得到 $Y(k)=W_N^{-km}X(k)$。

这说明，时域中信号的移位只影响其 DTFT 的相位，而不影响其幅频特性，这从物理意义上也是很好理解的。

（3）频域循环移位定理

如果 $X(k)=\mathrm{DFT}[\,x(n)]$，$0\leqslant k\leqslant N-1$，而 $Y(k)$ 是 $X(k)$ 的 l 位循环移位序例 $Y(k)=X((k+l)_N)R_N(k)$，则

$$y(n)=\mathrm{IDFT}[Y(k)]=W_N^{nl}x(n) \tag{3.6.9}$$

（3.6.9）式的证明方法与时域循环移位定理类似。因为本书不讨论时域中的复数序列，这个定理不做证明。

3.6.4　循环折叠性质

若把 N 点序列 $x(n)$进行折叠。那么 x 的下标$(-n)$将不再在 $0\leqslant n\leqslant N-1$ 区域内。要使它的下标仍在主值区内，必须要利用其隐含周期性对变量（$-n$）进行 N 求余运算，定义 $x(n)$循环折叠序列 $y(n)$为：

$$y(n)=x((-n)_N)=\begin{cases}x(0) & n=0\\ x(N-n) & 1\leqslant n\leqslant N-1\end{cases} \tag{3.6.10}$$

为了更形象化一些，想像序列 $x(n)$以反时针方向等间隔地放置在一个圆周上，那么 $x(-n)$ 可看成 $x(n)$沿着圆周顺时针放置。在 MATLAB 中，循环折叠可由 y=x(mod(-nx,N)+1)得到。注意 MATLAB 中的变量下标由 1 开始，而 mod 函数的结果却从零开始，因此上述语句中下标要加 1。循环折叠性质同样适合于频域。经循环折叠后，序列的 DFT 由下式给出：

$$Y(k)=\mathrm{DFT}[x((-n)_N)]=X^*((-k)_N)=\begin{cases}X(0) & k=0\\ X^*(N-k) & 1\leqslant k\leqslant N-1\end{cases} \tag{3.6.11}$$

即时域做循环折叠后的函数，其对应的DFT在频域也做循环折叠，并取共轭。其MATLAB表达式应为

$$Y=\text{conj}(X((\text{mod}(-kx,N)+1))$$

例 3.6.2：设 $x(n)=[3,4,5,6,8,9,10]$，循环长度 $N=10$，确定并画出 $x((-n)_{10})$。

解：首先要把这个序列放到循环的位置向量上去。

```
x = [3,4,5,6,8,9,10]; N = 10;              % 给出原始数据，假如 nx 从零开始
x=[x,zeros(1,N-length(x))]; nx=0:N-1;      % 将 x 的长度通过补零扩展到 N
y = x(mod(-nx,N)+1);                       % 把 x 循环折叠，求得 y
```

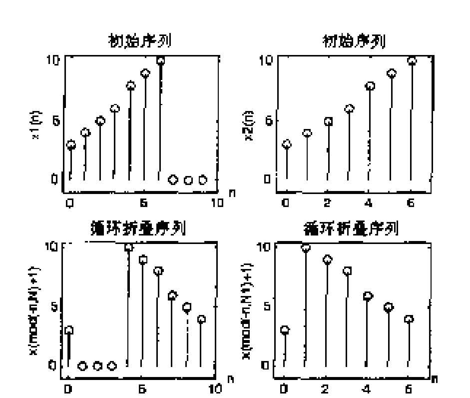


图 3.6.3　序列的循环折叠：左图 N=10，右图 N=7

再用 stem([0:N-1],x); stem([0:N-1],y)在主值区间画图，得到图 3.6.3(a)的两个子图。

如果循环长度为 N1=7，则将得出图 3.6.3(b)的两个子图，请读者自己思考编程。

3.6.5　DFT 的循环对称性

本章3.2.3节和3.2.4节中已详细讨论了实数和复数序列DTFT的对称性，那里的对称性是指关于坐标原点（零频率点）的对称性。DFT 也有类似的对称性，但在 DFT 中涉及的序列 $x(n)$ 及其离散傅里叶变换 $X(k)$的定义区间均为 0 到 N-1，所以循环对称性对时间序列是指关于 n=0 和 n=N/2 处的对称性，对频谱序列是关于数字频率为 0 和 pi 处的对称性。除了这一点之外，DTFT 和 DFT 的对称性是相仿的。实序列的 DTFT 和 DFT 的对比列于表 3.6.1 中，复序列的 DTFT 和 DFT 的对比列于表 3.6.2 中。

（1）实序列的循环对称性

定义于主值区间[0,N-1]的实序列 $x(n)$可以分解为循环偶序列 $x_e(n)$ 和循环奇序列 $x_o(n)$，

$$x(n)=x_e(n)+x_o(n)\qquad 0\leqslant n\leqslant N-1$$

其中：

$$x_e(n)=\frac{1}{2}[x(n)+x(-n)],\qquad x_o(n)=\frac{1}{2}[x(n)-x(-n)]\tag{3.6.12}$$

其中 $x(-n)$ 为 $x(n)$的循环折叠序列，所以也定义在主值区间 [0, N-1]上。循环偶序列 $x_e(n)$ 和奇序列 $x_o(n)$ 分别满足对称条件

$$x_e(n)=x_e[(-n)_N]=x_e(N-n),\quad 0\leqslant n\leqslant N-1\tag{3.6.13}$$

也满足反对称条件 $x_o(n)=-x_o[(-n)_N]=-x_o(N-n),\ 0\leqslant n\leqslant N-1$　　(3.6.14)

假如 N=8,对称条件意味着 $x(1)$=$x(7)$,$x(2)$=$x(6)$,$x(3)$=$x(5)$,而 $x(0)$和 $x(4)$则是对称中心；假如 N=7,对称条件意味着 $x(1)$=$x(6)$,$x(2)$=$x(5)$,$x(3)$=$x(4)$,而对称中心则位于 k=0 和 k=3.5。即在 k=3 和 k=4 两个样本点的中间。把序列 $x(n)$进行这种分解的目的是找到它的 DFT 中实部和虚部所对应的时间序列。

设 $\text{DFT}[\,x(n)\,]=X(k)=\text{Re}[\,X(k)\,]+\text{j}\cdot\text{Im}[X(k)]$，则

$$\text{DFT}[x_e(n)]=\text{Re}[X(k)]\tag{3.6.15}$$

$$\text{DFT}[x_o(n)]=\text{j}\cdot\text{Im}[\,X(k)\,]\tag{3.6.16}$$

证明（3.6.15）式：利用循环折叠性质公式（3.6.11），有

$\text{DFT}[x_e(n)] = \text{DFT}[0.5[x(n)+x(-n)]] = 0.5[X(k)+X^*(-k)] = \text{Re}[X(k)]$。

同理可证明(3.6.16)式，所以实序列中的偶序列 $x_e(n)$ 对应于 $x(n)$ 的离散傅里叶变换 $X(k)$ 的实部，而其奇序列 $x_o(n)$ 对应于 $x(n)$ 的离散傅里叶变换 $X(k)$ 的虚部。

例 3.6.3：设 N=8，定义在主值区间的序列 x=[ones(1,6)，zeros(1,2)]，试将其分解为偶对称序列和奇对称序列，求它们的 DFT 并验证本节所述的特性。

解：根据公式（3.6.12）～（3.6.16）可以编写 MATLAB 程序 hc364 如下：

```
N=8; x=[ones(1,6),zeros(1,2)];nx=0:N-1;       % 给定实序列x
xr =x(mod(-nx,N)+1);                           % 求x的循环折叠xr
xe=0.5*(x+xr); xo=0.5*(x-xr);                  % 分解为偶、奇序列xe,xo
X=fft(x,8), Xe=fft(xe,8), Xo=fft(xo,8)         % 求x,xe,xo的DFT
Error1=(max(abs(real(X)-Xe)))                  % 检验X的实部与DFT[xe]的误差
Error2=(max(abs(j*imag(X)-Xo)))                % 检验X的虚部与DFT[xo]的误差
用stem(nx,x) ,stem(nx,xr) ,stem(nx,xe) ,stem(nx,xo) 分别绘图
```

运行得出的结果为

```
X =[  6.0000     - 0.7071-1.7071i   1.0000-1.0000i   0.7071+0.2929i
          0       0.7071 - 0.2929i   1.0000 + 1.0000i  -0.7071 + 1.7071i ]
Xe =[  6.0000   -0.7071    1.0000    0.7071   0    0.7071    1.0000   -0.7071]
Xo =[ 0  - 1.7071i  - 1.0000i  + 0.2929i   0   - 0.2929i   1.0000i  + 1.7071i]
Error1≈0, Error2≈0
```

输入绘图命令可得如图 3.6.4 所示的序列波形。

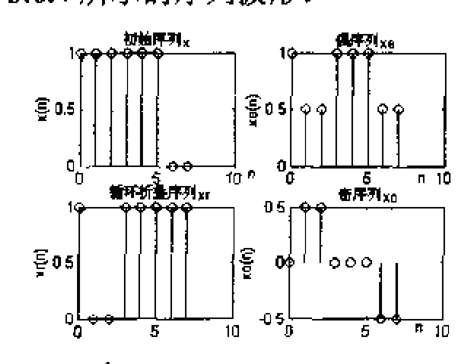


图 3.6.4　左，实序列及其循环折叠；右，分解成的偶奇序列

从 xe(n)和 xo(n)的序列波形，读者可进一步体会序列对 n=0 及 N/2 处对称的含义。

请读者思考在图 3.6.4 中：

a. $x(n)$的循环折叠序列 xr 为什么是这个形状？

b. $x(n)$各点之值是否等于 xe(n)+xo(n)？

c. xe(n)是否符合循环偶对称的定义？为什么？

d. xo(n)是否符合循环奇对称的定义？为什么？

从 X,Xe,Xo 的数据中，注意：

e. Xe(k)和 Xo(k)是否符合循环偶对称和循环奇对称的定义？为什么？

f. Xe(k)和 Xo(k)之和是否等于 $X(k)$?为什么？

g. Xe(k)和 Xo(k)是否分别对应于 $X(k)$的实部和虚部？为什么？

（2）实序列 DFT 循环对称特性的应用

从上面的推导中，可以更明确地归纳出以下三点结论。

a. 设 $x(n)$ 是长度为N的实序列，且 $X(k)=\mathrm{DFT}[x(n)]$，则 $X(k)$ 必循环共轭对称，即

$$X(k)=X^*((-k)_N)=X^*(N-k)] \qquad 0\leqslant k\leqslant N-1 \tag{3.6.17}$$

进一步，可以推论出：

DFT 的实部满足循环偶对称的条件为 $\mathrm{Re}[X(k)]=\mathrm{Re}\{X((-k)_N)\}$；

DFT 的虚部满足循环奇对称的条件为 $\mathrm{Im}[X(k)]=-\mathrm{Im}[X[(-k)_N]]$；

DFT 的模满足循环偶对称的条件为 $|X(k)|=|X((-k)_N)|$；

DFT 的相角满足循环奇对称的条件为 $\angle X(k)=-\angle X[(-k)_N]$。

由于 $X(k)$ 的虚部是奇序列，在 $k=0$ 处，必有 $\mathrm{Im}[X(0)]=0$，因而 $X(0)$ 必为实数。这表明了信号中的直流分量。同样，若 N 是偶数，则 $N/2$ 也是一整数。可得

$$X\left(\frac{N}{2}\right)=X^*\left(\left(\frac{N}{2}\right)_N\right)=X^*\left(\frac{N}{2}\right)$$

由于 $k=N/2$ 处的频率为 $\omega_{N/2}=(N/2)\cdot 2\pi/N=\pi$，它是数字奈奎斯特频率，所以此分量称为奈奎斯特分量，对实序列而言，它也是实数。

b. 如果 $x(n)$ 循环对称，即 $x(n)=x(N-n)$，则 $X(k)$ 为实数，且循环偶对称，即

$$X(k)=X(N-k) \tag{3.6.18}$$

c. 如果 $x(n)=-x(N-n)$，则 $X(k)$ 为纯虚数，且奇对称，即

$$X(k)=-X(N-k) \tag{3.6.19}$$

实际中经常需要对实序列求 DFT，利用上述对称性质，可减少 DFT 运算量。当 N=偶数时，只需计算前面 $N/2+1$ 点，而 N=奇数时，只需计算前面 $(N+1)/2$ 点。在采用 MATLAB 等计算软件时，运算量的节省并不重要，但却往往需要在绘图时减少图的面积。因此在绘制频谱曲线时，通常只画出频率为 0 到 π 的一段，而把 π～2π 省略了。因为按照对称性条件可以求得。

可以编写出把长度为 N 的循环实序列分解为偶奇序列的 MATLAB 函数程序 circevod.m 如下：

```
function [xe, xo] = circevod(x)
% 将循环实序列分解为循环偶和循环奇两部分
if any(imag(x) ~= 0)                          % 检查是否为实序列
  error('x 非实数序列')
end
N = length(x); n = 0:(N-1);
xe = 0.5*(x + x(mod(-n,N)+1));               % 构成偶序列
xo = 0.5*(x - x(mod(-n,N)+1));               % 构成奇序列
```

例 3.6.4：设 $x(n)=10\bullet(0.9)^n$，$0\leqslant n\leqslant 9$，分解并画出其循环偶奇序列，并讨论其傅里叶变换序列的对称性。

解：写出下列 MATLAB 程序。

```
n=0:9;x=10*(0.7).^n;N=length(x);
[xe, xo] = circevod(x);
Xe=fft(xe,N);Xo=fft(xo,N);
```

用六个子图分别画出 stem(n,xe),stem(n,real(Xe)), stem(n,imag(Xe)) 及 stem(n,xo), stem(n,real(Xo)), stem(n,imag(Xo))后得到图 3.6.5。

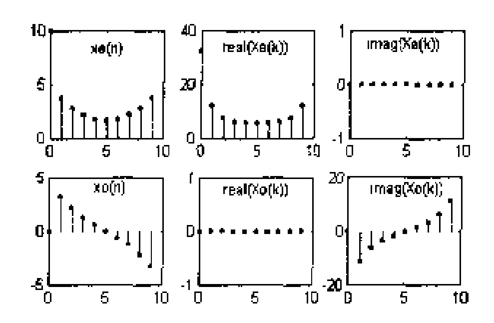


图 3.6.5　实数序列的偶和奇序列分别对应于其 DFT 的实部和虚部

从图上可以看出：

a. xe 具有循环对称性。对称中心为 n=5 处。表面上看最左的 xe(0)处，在最右的 xe(10)处没有和它对称的分量。但要知道，因为 N=10，xe 只定义在 n=0:9 上。到 xe(10)就转回到 xe(0)了。所以 xe(0)这一点是左右公用的，这样它的对称性就毫无问题了。

b. xo 具有循环反对称性。对称中心也为 n=0 和 n=5 处。根据反对称的定义，反对称中心处序列的值必须为零,即 xo(0)=xo(5)=0。

c. 从图上可看出，Xe(k)是对称序列 xe 的 DFT，它的虚部为零；Xo(k)是反对称序列 xe 的 DFT, 它的实部为零；由于 $X(k)$=Xe(k)+Xo(k),可知 Xe(k)是 $X(k)$ 的实部，Xo(k)是 $X(k)$ 的虚部。即时域的偶奇分量确实对应于其频域 DFT 的实部和虚部。

3.6.6　循环卷积特性

有限长序列 $x_1(n)$ 和 $x_2(n)$，长度分别为 N_1 和 N_2，$N=\max[N_1,N_2]$。设 $x_1(n)$ 和 $x_2(n)$ 的 N 点 DFT 分别为：$X_1(k)=\mathrm{DFT}[x_1(n)]$ 和 $X_2(k)=\mathrm{DFT}[x_2(n)]$。如果 $X(k)=X_1(k)\cdot X_2(n)$ 则

$$x(n)=\mathrm{IDFT}[X(k)]=\sum_{m=0}^{N-1}x_1(m)x_2((n-m)_N)R_N(n) \qquad (3.6.20)$$

这表示 $x(n)$ 为 $x_1(n)$ 与 $x_2(n)$ 的循环卷积。

证明：直接对（3.6.27）式两边进行 DFT，有

$$X(k)=\mathrm{DFT}[x(n)]=\sum_{n=0}^{N-1}\left[\sum_{m=0}^{N-1}x_1(m)x_2((n-m)_N)R_N(n)\right]W_N^{kn}$$

$$=\sum_{m=0}^{N-1}x_1(m)\sum_{n=0}^{N-1}x_2((n-m)_N)W_N^{kn}$$

令 $n-m=n'$，则有

$$X(k)=\sum_{m=0}^{N-1}x_1(m)\sum_{n'=-m}^{N-1-m}x_2((n')_N)W_N^{k(n'+m)}=\sum_{m=0}^{N-1}x_1(m)W_N^{km}\sum_{n'=-m}^{N-1-m}x_2((n')_N)W_N^{kn'}$$

上式中 $x_2((n')_N)W_N^{kn'}$ 是以 N 为周期的，所以可以取任一个周期求和而不影响结果。因此

$$X(k)=\sum_{m=0}^{N-1}x_1(m)W_N^{km}\cdot\sum_{n'=0}^{N-1}x_2(n')W_N^{kn'}=X_1(k)X_2(k),\quad 0\leqslant k\leqslant N-1 \qquad (3.6.21)$$

可见，如果（3.6.20）式成立，(3.6.21)式必然成立，命题得证。

（3.6.20）式称为时域循环卷积定理。在实现循环卷积过程中，要求对 $x_2(m)$循环折叠，循环移位，要注意两个长度为 N的序列的循环卷积长度仍为 N，显然与一般的线性卷积不同。因为线性卷积的长度为两个输入序列长度之和减 1。循环卷积的结果与循环长度的值 N紧密相关，因此记为：

$$x(n)=x_1(n)\ ⓝ\ x_2(n)=\sum_{m=0}^{N-1}x_1(m)x_2((n-m)_N)R_N(n) \qquad (3.6.22)$$

和线性卷积一样，由于 $X(k)=\mathrm{DFT}[x(n)]=X_1(k)\cdot X_2(k)=X_2(k)\cdot X_1(k)$，所以循环卷积满足交换律，如公式（3.6.23）所示。

$$x(n)=\mathrm{IDFT}[X(k)]=x_1(n)\ ⓝ\ x_2(n)=x_2(n)\ ⓝ\ x_1(n) \qquad (3.6.23)$$

由于 $x_1(n)$，$x_2(n)$，$X_1(k)$和 $X_2(k)$都是定义在主值区间上的循环序列，$x_1(n)$和 $x_2(n)$的移位、折叠和乘法也都应在 0～N-1 的循环区间上进行。为了编程的方便，可以先把输入序列延拓为周期序列，然后在线性位置向量上进行各种移位、折叠和乘法等运算，完成卷积后再取出结果在主值区间上的值。按这样的思路编出的程序比较冗长，留给读者作为课外的工作。

本书将采取建立循环序列，直接作循环卷积运算的方法。首先要把两个输入序列改成主值区间的循环序列，进行循环折叠和循环移位，然后求循环卷积。并把这个过程写成一个时域卷循环积函数 circonvt.m，以便直接调用。其中 y 为循环卷积的输出序列；$x1$，$x2$ 为输入序列；N 为循环长度；两输入序列的长度均应小于等于 N。

```
function y=circonv(x1,x2,N)
% 求序列的循环卷积，方法 y(n) = sum (x1(m)*x2(mod((n-m),N)+1))
% 检查 x1, x2 的长度的语句从略
x1=[x1,zeros(1,N-length(x1))];          % 将 x1 的长度扩展至 N
x2=[x2,zeros(1,N-length(x2))];          % 将 x2 的长度扩展至 N
m = [0: N-1];
x2m = x2(mod(-m,N)+1);                  % 将 x2 循环折叠
H = toeplitz(x2m,[0,x2(2:N)]);\         % 用 toeplitz 函数产生循环卷积矩阵
y = x1*H;                               % 用向量—矩阵乘法求卷积
```

本程序中采用了向量－矩阵相乘的方法来求卷积，循环卷积时与线性卷积的差别在于 H 的构成方法不同。在线性卷积中，H 的最左列中，h 才移进了一个元素，其余的就空着为零，而循环卷积时，$h(0)$上面的未移进来的元素要补到它的下方：每一列的右列要依次下移一位，在线性卷积中，最后一项移出矩阵后就不管了。而在循环卷积中，移出项必须补到第一行中，以构成循环。线性卷积矩阵有 Nx 行和 $L=Nx+Nh-1$ 列，而循环卷积矩阵为 N 行 N 列的方阵。比较下面两个卷积矩阵，就可以看出它们的主要区别。

线性卷积矩阵为
$$H=\begin{bmatrix} h(0) & h(1) & \cdots & h(L-1)\\ 0 & h(0) & \cdots & h(L-2)\\ \vdots & & \vdots & \vdots\\ 0 & 0 & \cdots & h(L-Nx)\end{bmatrix} \qquad (2.5.8)$$

循环卷积矩阵为
$$H_c=\begin{bmatrix} h(0) & h(1) & \cdots & h(N-1)\\ h(N-1) & h(0) & \cdots & h(N-2)\\ \vdots & \vdots & \ddots & \vdots\\ h(1) & h(2) & \cdots & h(0)\end{bmatrix} \qquad (3.6.24)$$

这两个矩阵的形式都属于 toeplitz 矩阵，所以用 toeplitz 函数来产生最为简便。读者在阅

读程序时要注意弄清 H=toeplitz(C,R)产生的结果，在 C 和 R 的长度相同($Nc=Nr=N$)时，是

$$H=\begin{bmatrix} C(1) & R(2) & \cdots & R(Nr) \\ C(Nc) & C(1) & \cdots & R(Nr-1) \\ \vdots & \vdots & \ddots & \vdots \\ C(2) & C(3) & C(Nc) & C(1) \end{bmatrix} \tag{3.6.25}$$

可以看到它是一个矩阵，$C(1)$占据了从左上到右下的对角线，其余各项都沿对角线相等布局。$R(1)$则被 $C(1)$所覆盖，所以程序的 toeplitz 语句中向量 R 的第一项可任设。

在线性卷积中，如果 x1 的长度为 Nx, h 的长度为 Nh, 线性卷积矩阵 H 的阶次是 Nx×(Nx+Nh-1)，输出 y 的长度是 Nx+Nh-1；而在循环卷积中， 输入 x 与 h 的长度都限定为 N（不够时补零，过长时截断），所以循环卷积矩阵 H_c 是 N×N 的方阵。输出 y 的长度当然也是 N。

例 3.6.5：设 $x_1(n)=\{1,2,3\}$ ，$x_2(n)=\{5,4,-3,-2,\}$ 计算 4 点循环卷积 $y(n)=x_1(n)④\ x_2(n)$。

解：分别在时域及频域中求解这个问题。在时域中采用循环卷积技术，包括手工计算和调用 MATLAB 函数程序；而在频域中采用 DFT。这样，一共有如下三种方法来比较[注意 $x_1(n)$ 为三点序列，进行先进卷积之前必须在其尾部填一个零，使其成为四点序列]：

（1）时域手工方法

四点循环卷积的每一点由下式给出：

$$y(n)=x_1(n)④x_2(n)=\sum_{m=0}^{3}x_1(m)x_2((n-m)_4) \qquad n=0,1,2,3$$

对每一个 n，要产生一循环移位序列，将它的样本逐个与 $x_1(m)$相乘，然后求和，得到 n 值的循环卷积值，在 $0\leqslant n\leqslant 3$ 上重复此过程。由

$x_1(m)=[1,2,3,0]$，$x_2(m)=[5,4,-3,-2]$，　　$x_2(-m)=[5,-2,-3,\ 4]$，

依次将 n=0,1,2,3 代入进行计算，可得

$$y(0)=\sum_{m=0}^{3}x_1(m)x_2((0-m)_4)=\sum_{m=0}^{3}\big[[1,2,3,0]\bullet[5,-2,-3,4]\big]=5-4-9+0=-8$$

$$y(1)=\sum_{m=0}^{3}x_1(m)x_2((1-m)_4)=\sum_{m=0}^{3}\big[[1,2,3,0]\bullet[4,5,-2,-3]\big]=4+10-6+0=8$$

$$y(2)=\sum_{m=0}^{3}x_1(m)x_2((2-m)_4)=\sum_{m=0}^{3}\big[[1,2,3,0]\bullet[-3,4,5,-2]\big]=-3+8+15=20$$

$$y(3)=\sum_{m=0}^{3}x_1(m)x_2((3-m)_4)=\sum_{m=0}^{3}\big[[1,2,3,0]\bullet[-2,-3,4,5]\big]=-2-6+12+0=4$$

由此得到循环卷积的结果为 y=[-8, 8, 20, 4]。

（2）调用 MATLAB 函数程序的方法

```
x1=[1,2,3,0];x2=[5,4,-3,-2];
y=circonv(x1,x2,4)
```

结果为 `y = -8    8    20    4`。

（3）用频域 DFT 相乘再求反变换方法

$$X_1(k)=\mathrm{DFT}[x_1(n)]=\mathrm{e}^{-\mathrm{j}0}+2\mathrm{e}^{-\mathrm{j}\pi k/2}+3\mathrm{e}^{-\mathrm{j}k\pi}+0\cdot\mathrm{e}^{-\mathrm{j}3k\pi/2}$$

$X_2(k)=\mathrm{DFT}[x_2(n)]=5\cdot e^{-j0}+4\cdot e^{-j\pi k/2}-3\cdot e^{-jk\pi}-2\cdot e^{-j3k\pi/2}$

将 k=0,1,2,3 代入，可得

$X_1=[6.0000 \quad -2.0000-2.0000i \quad 2.0000 \quad -2.0000+2.0000i]$

$X_2=[4.0000 \quad 8.0000-6.0000i \quad 0 \quad 8.0000+6.0000i]$

将 X_1 及 X_2 按 k 逐项相乘，得到

$X=[24.0000 \quad -28.0000-4.0000i \quad 0 \quad -28.0000+4.0000i]$

卷积结果为 X 的离散傅里叶反变换 $y(n)$=IDFT[$X(k)$]

$$y(n)=\mathrm{IDFT}[X(k)]=\frac{1}{4}\left(24\cdot e^{j0}-(28+j4)e^{jn\pi/2}+0\cdot e^{jn\pi}-(28-j4)\cdot e^{j3n\pi/2}\right)$$

设 n=0,1,2,3，可算出 y=[-8,8,20,4]。

结果当然是一样的。但是用手工来计算频域的问题，麻烦的是要遇到复数。为了搞清概念，可以把简单的问题算一算，实际上 N 稍大一些，就必须用计算机。用 MATLAB 来做此频域计算，只要以下几条简单语句就行了。

```
X1=fft(x1,4);X2=fft(x2,4);Y=X1.*X2;y=ifft(Y,4)
```

注意：在此题中 $x1$ 与 $x2$ 长度不同，而且我们要求的是长度为 4 的循环卷积，所以 fft 和 ifft 中的第二个输入变元 N 是不能省略的，不然计算就会出错。

循环长度 N 是很重要的参数，比如把上题中的 N 改成 8，运行下列程序：

```
X1=fft(x1,8);X2=fft(x2,8);Y=X1.*X2;y=ifft(Y,8)
```

得到的结果为 `y = [5  14  20   4 -13  -6   0   0]`。

与上面的四点循环卷积结果完全不同。实际上，这是线性卷积的结果。当 $N \geq Nh+Nx-1$ 时，循环卷积的首尾不会重叠，它就和线性卷积相同了。这可以用 y=conv(x1,x2)来验证。

在实际信号处理中，要计算的通常是线性卷积。为了快速地计算线性卷积，要借助于快速傅里叶变换，于是就会得到循环卷积的结果，如果点数 N 不够大，循环卷积会把卷积输出的头尾部分混叠到一起，造成错误。所以，必须把循环长度 N 取得比两个序列长度之和大，才能保证循环卷积等于线性卷积。

DFT 在频域的卷积同样有循环性。频域卷积对应于时域序列的相乘，即如果 $x(n)=x_1(n)x_2(n)$，则它们的 DFT 满足循环卷积关系，如（3.6.33）所示。

$$X(k)=\frac{1}{N}X_1(k) \circledN X_2(k)=\frac{1}{N}\sum_{j=0}^{N-1}X_1(j)X_2((k-j)_N)R_N(k) \tag{3.6.26}$$

因为序列的相乘在时域通常称为调制，因此这一关系称为相乘特性，也称为调制特性。可以从表 3.6.2 中查到。

3.6.7 帕瑟瓦尔定理

在 DFT 领域，同样有帕瑟瓦尔（Parseval）定理，可用于计算频域的能量

$$E=\sum_{n=0}^{N-1}|x(n)|^2=\frac{1}{N}\sum_{n=0}^{N-1}|X(k)|^2 \tag{3.6.27}$$

$\dfrac{|X(k)|^2}{N}$ 叫做有限长序列的能量频谱。类似地，对于周期信号，$\left|\dfrac{X(k)}{N}\right|^2$ 叫做功率频谱。

表 3.6.1　实数序列 DTFT 及 DFT 的对称关系

序　列	DTFT	DFT
$x(n)$	$X(e^{j\omega})=X_{re}(e^{j\omega})+jX_{im}(e^{j\omega})$	$X(k)=\mathrm{Re}[X(k)]+j\mathrm{Im}[X(k)]$
$x_{ev}(n)$	$X_{re}(e^{j\omega})$	$\mathrm{Re}[X(k)]$
$x_{od}(n)$	$jX_{im}(e^{j\omega})$	$j\mathrm{Im}[X(k)]$
对称关系	$X(e^{j\omega})=X^*(e^{-j\omega})$	$X(k)=X^*((-k)_N)$
	$X_{re}(e^{j\omega})=X_{re}(e^{-j\omega})$	$\mathrm{Re}[X(k)]=\mathrm{Re}[X((-k)_N)]$
	$X_{im}(e^{j\omega})=-X_{im}(e^{-j\omega})$	$\mathrm{Im}[X(k)]=-\mathrm{Im}[X((-k)_N)]$
	$\lvert X(e^{j\omega})\rvert=\lvert X(e^{-j\omega})\rvert$ $\arg[X(e^{j\omega})]=-\arg[X(e^{-j\omega})]$	$\lvert X(k)\rvert=X\lvert((-k)_N)\rvert$ $\arg[X(k)]=-\arg[X((-k)_N)]$

表 3.6.2　序列傅里叶变换和离散傅里叶变换的特性

变换 \ 特性	DTFT 的特性		DFT 的特性	
	时　域	频　域	时　域	频　域
	$g(n)$ $h(n)$	$G(e^{j\omega})$ $H(e^{j\omega})$	$g(n)$ $h(n)$	$G(k)$ $H(k)$
线性性	$\alpha g(n)+\beta h(n)$	$\alpha G(e^{j\omega})+\beta H(e^{j\omega})$	$\alpha g(n)+\beta h(n)$	$\alpha G(k)+\beta H(k)$
时移性	$g[n-n_0]$	$e^{-j\omega n}0G(e^{j\omega})$	$g[(k-k_0)_N]$	$W_N^{kn_0}G(k)$
频移性	$e^{j\omega_0 n}g(n)$	$G\left(e^{j(\omega-\omega_0)}\right)$	$W_N^{-kn_0}g(n)$	$G((k-k_0)_N)$
频域导数	$ng(n)$	$j\dfrac{dG(e^{j\omega})}{d\omega}$		
卷积	$g(n)\otimes h(n)$	$G(e^{j\omega})H(e^{j\omega})$	$\sum\limits_{m=0}^{N-1}g(m)h((n-m)_N)$	$G(k)H(k)$
调制	$g(n)h(n)$	$\dfrac{1}{2\pi}\int_{-\pi}^{\pi}G(e^{j\theta})H(e^{j(\omega-\theta)})d\theta$	$g(n)h(n)$	$\dfrac{1}{N}\sum\limits_{m=0}^{N-1}G(m)H((k-m)_N)$
能量关系	$\sum\limits_{n=-\infty}^{\infty}g(n)h^*(n)=\dfrac{1}{2\pi}\int_{-\pi}^{\pi}G(e^{j\omega})H^*(e^{j\omega})d\omega$		$\sum\limits_{n=0}^{N-1}\lvert x(n)\rvert^2=\dfrac{1}{N}\sum\limits_{k=0}^{N-1}\lvert X(k)\rvert^2$	

3.7　用 DFT 计算线性卷积

线性系统中最重要的一个运算是线性卷积，但在时域中进行线性卷积的效率是很低的。另一方面，DFT 是在频域中实现线性卷积的实用方法。它有进行高效计算的方法 FFT，但 DFT 的结果是循环卷积而不是线性卷积，通常这不是所希望的。怎样用 DFT 执行线性卷积？这就是本节要讨论的问题。

3.7.1 循环卷积与线性卷积

设 $x_1(n)$为 N_1 点序列，$x_2(n)$为 N_2 点序列，$y(n)$定义为 $x_1(n)$和 $x_2(n)$的线性卷积，即：

$$y(n)=x_1(n)\otimes x_2(n)=\sum_{m=-\infty}^{\infty}x_1(m)x_2(n-m)=\sum_{m=0}^{N_1-1}x_1(m)x_2(n-m) \tag{3.7.1}$$

从第 2 章知道，$y(n)$为（N_1+N_2-1）点序列。

根据循环卷积定理，两个循环长度分别为 N_1 和 N_2 的序列的卷积，如果选择一个公共的循环长度 $N=\max$（N_1，N_2），就可以按图 3.7.1 的框图求它们的循环卷积，先做两个 N 点的 DFT，相乘后，再做一个 N 点的 IDFT 来完成。由于 DFT 有快速计算的方法，计算 N 点循环卷积 $x_1(n)$ Ⓝ $x_2(n)$，将比时域中直接计算卷积的速度高几个数量级。问题是，这样计算得到的 N 点序列与线性卷积 $y(n)$的结果是否相同？有没有误差？

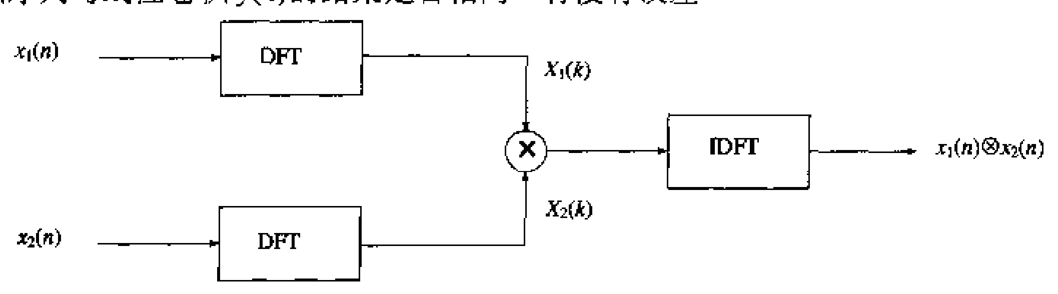


图 3.7.1　用 DFT 实现循环卷积的框图

3.6 节例的题已给出一个启示，如果选择 $N=N_1+N_2-1$，进行 N 点循环卷积，就有可能避免首尾混叠。下面来证明这一点。设 $N=N_1+N_2-1$，把 $x_1(n)$和 $x_2(n)$看成两个 N 点序列，定义 $x_3(n)$为 N 点循环卷积。可以写出

$$x_3(n)=x_1(n) \text{Ⓝ} x_2(n)=\left[\sum_{m=0}^{N-1}x_1(m)x_2((n-m)_N)\right]R_N(n) \tag{3.7.2}$$

将 $x_2(n)$的周期延拓按(3.6.1)式写成平移求和形式为

$x_2((n-m)_N)=\sum_{r=-\infty}^{\infty}x_2(n-m-rN)$，代入 (3.7.2)式，可以求得

$$x_3(n)=\left[\sum_{m=0}^{N-1}x_1(m)\sum_{r=-\infty}^{\infty}x_2(n-m-rN)\right]R_N(n)$$

$$=\left[\underbrace{\sum_{r=-\infty}^{\infty}\sum_{m=0}^{N-1}x_1(m)x_2(n-m-rN)}_{y(n-rN)}\right]R_N(n)=\left[\sum_{r=-\infty}^{\infty}y(n-rN)\right]R_N(n) \tag{3.7.3}$$

(3.7.3)式说明，循环卷积 $x_3(n)$是线性卷积 $y(n)$的一种混叠形式。如果

$$N\geqslant (N_1+N_2-1) \tag{3.7.4}$$

即 N 不小于 y 的长度，则(3.7.3)式中求和号中不同 r 所对应的 y 不会重叠，这就意味着在时域中不存在混叠。如果通过填充适当数目的零使 $x_1(n)$和 $x_2(n)$成为长度等于 $N\geqslant N_1+N_2-1$ 的序列，那么循环卷积与线性卷积相等。

$$x_3(n)=y(n) \qquad 0\leqslant n\leqslant N-1$$

例 3.7.1：设 $x_1(n)$和 $x_2(n)$是两个四点序列：

$x_1(n)=\{1,2,2,1\}$，　$x_2(n)=\{1,-1,1,-1\}$

a. 确定它们的线性卷积 $y(n)$。

b. 计算循环卷积 $x_3(n)$，使得它与 $y(n)$相等。

解：用 MATLAB 来解决这个问题。线性卷积可以调用 conv 函数，循环卷积也可调用 circonvt 函数，关键是循环卷积的长度要正确选择，必须使 $N\geqslant N_1+N_2-1=4+4-1=7$，选择 $N=7$，可以编出程序 hc371 如下：

```
x1=[1,2,2,1];x2=[1,-1,1,-1];
y=conv(x1,x2)                    % 线性卷积
x3 = circonvt(x1,x2,7)           % 循环卷积，长度为 7
```

程序运行的结果为

```
y =    1    1    1    0   -1   -1   -1
x3 =   1    1    1    0   -1   -1   -1
```

因此 $y(n)=x_3(n)$。

本例说明，为了用 DFT 做线性卷积，必须选择足够大的 N。

误差分析：在有些情况下 N 可能达不到(3.7.4)式，尤其当 N 很大而存储空间却有限时。如果不得不选择比所需要的 N 小的值进行循环卷积，计算的结果就不等于线性卷积，产生了误差。在实际中有时需要计算此误差。显然，从卷积的定义来说，至少要把 N 选得比每个序列长，即至少应满足 $N\geqslant\max(N_1, N_2)$。设

$$\max(N_1, N_2)\leqslant N<(N_1+N_2-1) \tag{3.7.5}$$

由(3.7.3)式（注意 $y(n)$就是求和号下 r=0 项），可以得出误差为：

$$e(n)=x_3(n)-y(n)=\left[\sum_{r\neq 0}y(n-rN)\right]R_N(n) \tag{3.7.6 a}$$

因为 $N\geqslant\max(N_1, N_2)$，上面的求和式中只剩下对应于 $r=\pm1$ 的两项。因此

$$e(n)=\left[y(n-N)+y(n+N)\right]R_N(n)$$

一般来讲，$x_1(n)$和 $x_2(n)$为因果序列，因此 $x_3(n)$也为因果序列，即：$y(n-N)=0$，$0\leqslant n\leqslant N-1$。因此

$$e(n)=y(n+N)\qquad 0\leqslant n\leqslant N-1 \tag{3.7.6 b}$$

这是一个简单但很重要的关系。它说明当 $\max(N_1, N_2)\leqslant N<(N_1+N_2-1)$时，样本 n 处的误差与线性卷积的第 $n+N$ 处的样本相等。在 n 较小的区域，$n+N$ 可能还小于(N_1+N_2-1)，因此 $y(n+N)$还有值，即存在误差。到第(N_1+N_2-1)个样本后，线性卷积结果为零。这说明循环卷积中只有头几个样本存在误差，剩下的则为正确的线性卷积样本。

例 3.7.2：考察上例中的序列 $x_1(n)$和 $x_2(n)$，计算当 N=6，5，4 时的循环卷积。在每种情形下验证误差关系。

解：显然，线性卷积 $y(n)$仍相同：$y(n)=\{1,1,-1,-2,-1,1,1\}$。当 N=6 时，得到六点序列：

$$x_3(n)=x_1(n)\,⑥\,x_2(n)=[2,1,-1,-2,-1,1],$$

因此 $e(n)=\{1,1,-1,-2,-1,1\}-\{1,1,-1,-2,-1,1\}=\{1,0,0,0,0,0\}=y(n+6)$　$0\leqslant n\leqslant 5$

与预想的相同。当 N=5,4 时，用下列 MATLAB 语句来算。

```
for N=[6,5,4]
   x3 = circonv(x1,x2,N)
   e=x3-y(1:N)
end
```

结果是：N=5 时，$x_3(n)=x_1(n)$ ⑤ $x_2(n)=[2, 2, -1, -2, -1]$，$e(n)=[1,1,0,0,0]=y([0:4]+5)=y([5:9])$。

当 N=4 时：$x_3(n)=x_1(n)$ ④ $x_2(n)=[0, 2, 0, -2]$，

$$e(n)=[0,2,0,-2]-[1,1,-1,-2]=[-1,1,1,0]=y([4:7])$$

结论是，选择 $N=\max(N_1, N_2)$ 作循环卷积时，前（$M-1$）个样本存在误差（不同于线性卷积），其中 $M=\min(N_1, N_2)$。此结果在用分段处理法实现长卷积时是非常有用的。

3.7.2 分段卷积

当要对一个连续接收的输入序列（像从麦克风得到的语音信号）进行滤波时，出于实用目的，可将此信号看做无限长序列。若想用 FIR 滤波器来实现滤波，它的 DFT 是用线性卷积计算的，那就必须计算一个巨大的 DFT，一般来讲，这是不切实际的；而且，需要所有的样本输入后，才能得到输出样本，这不符合实时处理的要求。因此必须把无限长度的序列分成较小的几部分（或段），用 DFT 处理每一部分，然后把每段的输出装配起来，成为输出序列。这个过程叫做分段卷积（或分段处理）。

假设把序列 $x(n)$分成多段 N 点序列，滤波器的脉冲响应为 M 点序列，$M \ll N$。用输入段和脉冲响应之间的 N 点循环卷积产生该段的输出序列。由上面的结论，可知其中前（$M-1$）个样本不是正确的输出值。若将 $x(n)$简单地分成互不重叠的各段，则所得的输出序列会有不正确样本区间存在。为了纠正这个问题，可以使 $x(n)$分成相互重叠（$M-1$）个样本的许多段，在卷积的结果中抛弃前面（$M-1$）个样本，保留后面的（$N-M+1$）个输出样本。最后把正确的输出样本连接成一个序列。为了修正第一段中的前（$M-1$）个样本，把第一个输入段的前（$M-1$）个样本置为零。此过程叫做分段卷积的重叠保留法(overlap/save)。显然，此方法适用于当 $N \gg M$ 的情况。

重叠保留法的计算原理可由下面的图 3.7.2 来形象地说明。其中：

(a)图说明如何把输入序列 $x(n)$分段，图上表示了三段，每段长度为 N=11，互相重叠部分的长度为 4，这是因为其脉冲响应的 $h(n)$长度为 M=5 的缘故，于是各段之间的间距是 11－4＝7。

(b)图表示每段卷积后的输出，因为是循环卷积，输出长度就等于循环长度，也即是输入长度 11。于是各输出段之间也有四个重叠样本，属于每段前面的是有误差的样本，应该去除或“拒绝”。

(c)图则表示把错误样本去除并将各输出段连接起来的结果，这也就是最后的线性卷积输出 $y(n)$。

下面将用简单的例子演示这一过程。

例 3.7.3：设 $x(n)=(n+1), 0\leqslant n\leqslant 9$, $h(n)=\{1,0,-1\}$,按 N＝6 用重叠保留法计算线性卷积 $y(n)=x(n)\otimes h(n)$

解：由于 M＝3，必须使每一段与前一段相重两个样本，$x(n)$为 10 点序列，需要在开头加(M-1)=2 个零，因而 N=6,可分为三段：

$x_1(n)=\{0, 0, x(1), x(2), x(3), x(4)\}=\{0, 0, 1, 2, 3, 4\}$

$x_2(n)=\{x(3), x(4), x(5), x(6), x(7), x(8)\}=\{3, 4, 5, 6, 7, 8\}$

$x_3(n)=\{x(7), x(8), x(9), x(10), 0, 0\}=\{7, 8, 9, 10, 0, 0\}$

因为 $x(n)$在 n=9 以外无值，因此在 $x_3(n)$最后必须填两个零。现在计算每一段与 $h(n)$的 L=6 点循环卷积。

$y_1=x_1(n)$ ⑥ $h(n)=[-3, -4, 1, 2, 2, 2]$

$y_2=x_2(n)$ ⑥ $h(n)=[-4, -4, 2, 2, 2, 2]$

$y_3=x_3(n)$ ⑥ $h(n)=[7, 8, 2, 2, -9, -10]$

注意丢弃各段的前两个样本，把 y1,y2,y3 装配成输出 $y(n)$

$y(n)=\{1, 2, 2, 2, 2, 2, 2, 2, 2, 2, -9, -10\}$

而这也就是它的线性卷积 conv(x,h)。

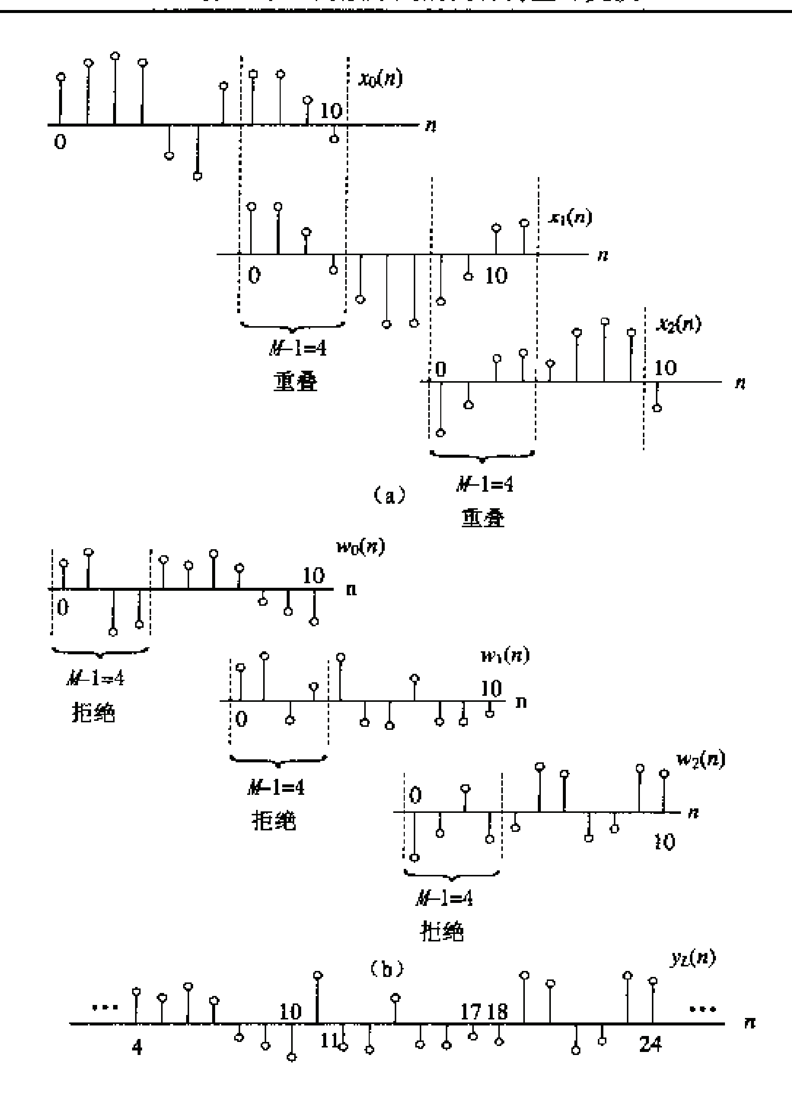


图 3.7.2　重叠保留法分段卷积的计算流程图

3.7.3　分段卷积的 MATLAB 函数

以上述例子为指导，可以设计一个 MATLAB 重叠保留法卷积函数 ovrlpsav.m，来实现长序列 $x(n)$ 与短序列 $h(n)$ 的卷积。此函数中关键的一步是为找到一个为分段定出下标的方法。在分段处理前，先将 $x(n)$ 前面放置（$M-1$）个零样本。扩大后的 x 序列设为 xz = [zeros(1，M-1), x];

设 $L=N-M+1$ 为每一段的有效数据的长度，则第 k 段 $x_k(n)$，$0\leqslant n\leqslant N-1$ 的数据为：

$$x_k(n)=xz(m);\quad kL\leqslant m\leqslant kL+N-1,\qquad k\geqslant 0,\ 0\leqslant n\leqslant N-1$$

所分的段数为：

$$K=\left\lfloor \frac{N_x+M-2}{L}\right\rfloor+1 \tag{3.7.4}$$

其中：N_x 是 $x(n)$ 的长度，$\lfloor\bullet\rfloor$ 是向下取整运算，如 $\lfloor 3.14\rfloor=3$，在 MATLAB 中与 floor 函数等价。在上例中，$Nx=10$，$M=3$，取 $N=6$，故 $L=6-3+1=4$，故 $K=\left\lfloor \frac{N_x+M-2}{L}\right\rfloor+1=\lfloor 2.75\rfloor+1=3$。

现在可用早先探讨的 circonvt 函数计算每一段与 $h(n)$ 的循环卷积为 $y_k(n)=x_k(n)⑥h(n)$。最后，丢弃每一个 $y_k(n)$ 中的前（$M-1$）个样本，连接剩下的样本，得到线性卷积 $y(n)$。这个过程由下面的 ovrlpsav 函数描述。

```
function [y] = ovrlpsav(x,h,N)            % 用重叠保留法做分段卷积
Lx = length(x); M = length(h);            % 输入数据 x 长度及脉冲响应 h 长度
M1 = M-1; L = N-M1;                       % 各段搭接长度 M1，有效数据长度 L
h = [h, zeros(1,N-M)];                    % 将 h 延长至循环长度 N
x = [zeros(1,M1), x, zeros(1,N-1)];       % 把 x 前面加上 (M-1) 个零
K = floor((Lx+M1-1)/(L))+1;               % 段数
Y = zeros(K+1,N);                         % 把 K+1×N 阶输出矩阵 Y 初始化
for k=0:K-1                               % 各段进行卷积
    xk = x(k*L+1:k*L+N);
    Y(k+1,:) = circonvt(xk,h,N);
end
Y = Y(:,M:N)';                    % Y 中各行均去掉前(M-1)个样本，转置后构成新的 Y
y = (Y(:))';                      % 装成单列向量再转置成行向量输出
```

应该指出，这里给出的 ovrlpsav 函数没有采用 FFT，所以它的计算效率不是很高的。它只是给出了一种编程的思路。放进 FFT 也很容易，只要把 for 循环中的语句按图 3.7.1 的框图编程即可。

另一种分段卷积的方法是重叠相加(overlap/add)法。在此方法中，把输入序列 $x(n)$分为互不重叠长度为 N 的 K 段，每段分别与长度为 M 脉冲响应 $h(n)$按 $N+M-1$ 的长度作循环卷积，这样所得的 K 个输出段 $y_k(n)$（k=1,1,…,K）的长度都是 $N+M-1$。各段的长度大于间隔，所以各段之间有 $M-1$ 个重叠的样本。把这 K 个输出段相加时把重叠的样本叠加在一起，就形成了最终的输出。MATLAB 中信号处理工具箱中的函数 fftfilt.m 就是按这个方法编写的。其中在求各段卷积时采用了快速傅里叶变换 FFT，因此其求卷积的速度很快。求卷积就是求滤波器输出，所以这个函数命名为 fftfilt。它适合于输入序列很长而脉冲响应较短的情况。它有两种调用格式：

（1）y=fftfilt(h, x)

（2）y=fftfilt(h, x, r)

其中 h 是系统或滤波器的（有限长）脉冲响应，x 是输入序列。在第一种调用格式中，程序自动把输入分成每段 512 个样本。并按 512 点（如果 $h(n)$的长度比 512 长，则按 $h(n)$的长度）FFT 进行各段的卷积运算。在第二种调用格式中，r 是用户指定的 FFT 长度，当然它必须比 $h(n)$的长度大，而输入 x 就得按这个长度分段。

例 3.7.4：利用例 3.7.3 中的序列，验证 ovrlpsav 和 fftfil t 函数。

解：MATLAB 程序为

```
n = 0:9; x = n+1; h = [1,0,-1]; N = 6;
y1 = ovrlpsav(x,h,N)
y2 = fftfilt(h,x)
```

运行结果为

```
y1 =    1     2     2     2     2     2     2     2     2     2    -9   -10
y2 =  1.00  2.00  2.00  2.00  2.00  2.00  2.00  2.00  2.00  2.00
```

可以看出，这两者之间的不同在于 fftfilt 函数产生的结果长度为 10，而我们自编的 ovrlpsav 函数的结果长度为 12。产生误差的就是这最后两个样本。原因在于 fftfilt 函数是专门对付很长的输入，它的分段长度就是 512，而要把好多个 512 长的样本段连接起来，所以输出的卷积长度要等于输入长度。现在本题杀鸡用了牛刀，输入只有 10 个样本，输出也就限于 10 个，函数 fftfilt 等着还有新输入时可以衔接。而我们自编的函数里却不是这样处理的，输入长度不合适，就随便给它最后补了两个零，所以导致在尾巴上两个样本的差错。

复习思考题

3.1 模拟频率的单位有哪几种？数字频率的单位是什么？它与几种模拟频率的关系是什么？为什么数字频率的单位中没有时间量纲？它的物理意义是什么？

3.2 数字频率和模拟频率的取值范围有什么不同？为什么把取样频率之半称为折叠频率？

3.3 模拟信号的频谱和数字信号的频谱各是怎样定义的？

3.4 模拟信号的傅里叶变换和傅里叶级数各是怎么定义的？对信号有什么限制条件？两者各用在什么场合？相互有什么关系？

3.5 序列信号的离散时间傅里叶变换(DTFT)定义是什么？对信号有什么限制条件？与模拟信号的傅里叶变换有什么异同？

3.6 归纳出 DTFT 的主要特性，并与连续信号傅里叶变换(CTFT)对比。

3.7 如何由 DTFT 的定义得出它的 MATLAB 计算公式(3.2.30 a)：X=x*exp(j*dw*n'*k)。在这个公式中的 dw,n 和 k 各应该如何确定。

3.8 离散时间系统的频率特性是如何定义的？它与系统的差分方程是什么关系？MATLAB 中用什么函数来计算频率特性？调用时必须知道哪些参数？

3.9 若序列重复 K 个周期，则它的 DTFT 与单个周期时有何不同？若 K 趋向于无穷大，则计算结果有什么特点？计算技术上又有什么问题？

3.10 为什么周期序列不能用 DTFT 而要用 DFS（离散傅里叶级数）来表示其频谱？能不能用 DFT（离散傅里叶变换）来表示其频谱？为什么？

3.11 DFT 的定义是什么？DFT 与 DTFT 有哪些不同？有哪些关系？即怎样由前者求出后者或由后者求出前者？DFT 与 DFS 有哪些不同？有哪些关系？

3.12 归纳 DFT 的主要特性，并与 DTFT 对比。

3.13 何谓序列的循环移位？如何解释下列表达式？

$$\tilde{x}(n) \cdot R_N(n) = x((n)_N) \cdot R_N(n) = \sum_{r=-\infty}^{\infty} x(n+mN) \cdot R_N(n)$$

3.14 何谓序列的循环折叠？如何解释 $\tilde{x}(-n) \cdot R_N(n) = x((N-n)_N) \cdot R_N(n)$？

3.15 何谓循环对称性？实序列的 DFT 的循环对称性用什么数学关系反映？

3.16 何谓循环卷积？它和线性卷积有什么区别？在什么条件下两者没有区别？

3.17 长度分别为 $L1$ 和 $L2$ 的两个序列 $x1(n)$ 和 $x2(n)$ 进行 N 点循环卷积。如果 N=max ($L1$,$L2$)，M=min ($L1$,$L2$)，问循环卷积的输出长度是多少？其中有多少个样本是与线性卷积不同的？排列在什么位置上？

3.18 试从线性卷积矩阵和循环卷积矩阵的区别，说明两种卷积的不同点。

3.19 什么叫分段卷积？什么叫重叠保留法？如何用重叠保留法进行分段卷积？

3.20 解释本书中的循环卷积子程序 circonv。

习题

3.1 下列序列中，哪一个是周期序列？求出它们的周期（样本数）。

(a) $\sin(2.5n)$　　(b) $\sin(4.8\pi n)$　　(c) $e^{j2.8\pi n}$　　(d) $\cos(7\pi n/17)$

(e) $3\sin(1.2\pi n+0.65\pi)+4\sin(0.8\pi n)-\cos(0.8\pi n)$

3.2 设题 3.1 中信号的采样频率为 2 Hz，试求出这些信号的模拟频率；如果它们的采样周期为 0.1s，则这些信号的模拟频率为多少？

3.3 用解析法求出以下各序列的 DTFT，用 MATLAB 画出 $X(e^{j\omega})$ 的幅值和相角曲线。

(a) $x(n)=5(0.8)^n\mu(n)$　　(b) $x(n)=2(0.95)^{n+2}\mu(n-2)$

(c) $x(n)=n(0.6)^n\mu(n)$　　(d) $x(n)=5(-0.8)^n\cos(0.1\pi n)\mu(n)$

(e) $x(n)=(n+1)(-0.8)^{n-2}\mu(n-2)$

3.4 对以下各个序列,求出其 DTFT $X(e^{j\omega})$。在频率范围[$-\pi,\pi$]内画出其幅值和相位曲线，并进行讨论。

(a) $x(n)$=[-1, 1, 3, 5, 3, 1, -1]　　(b) $x(n)$=[-1, 1, 3, 5, 5, 3, 1, -1]

(c) $x(n)$=[-1, 1, 3, 5, 0, -5, -3, -1, -1]　　(d) $x(n)$=[-1, 1, 3, 5, -5, -3, -1, -1]

3.5 写出一个 MATLAB 函数以计算一个有限长序列的 DTFT。利用这个函数来计算以后习题中遇到的 DTFT。该函数的格式应为：

```
function [X] = mydtft(x,w,n)
% 计算离散时间傅里叶变换
% -----------------------------------
% [X] = mydtft(x,w,n)
% X = 在w频率点上的DTFT数组 （行向量）
% x = 有限长度序列（行向量）
% n = 样本位置向量（行向量，缺省时取为x的长度）
% w = 频率点位置向量（行向量）
```

3.6 一对称上升余弦脉冲表达式为：$C_N(n)=\left[0.5+0.5\cos\left(\frac{2\pi n}{N}\right)\right]R_N(n)$，求出当 N=5, 25, 100 时的 DTFT，对它乘以因子使 $X(e^{j0})=1$。在[$-\pi,\pi$]区间画出归一化的 DTFT，研究这些曲线并评论它们随 N 变化的关系。

3.7 设 $X(e^{j\omega})$ 是实序列 $x(n)$的 DTFT，求用 $x(n)$表示的 $Y(e^{j\omega})=X(e^{j3\omega})$ 的 IDTFT $y(n)$。

3.8 设 $X(e^{j\omega})$ 是实序列 $x(n)$的 DTFT，求 $Y(e^{j\omega})=0.5[X(e^{j\omega/2})+X(e^{-j\omega/2})$ 的 IDTFT $y(n)$，用 $x(n)$表示。

3.9 一对称矩形脉冲序列

$$R(n)=\begin{cases}1 & -N\leqslant n\leqslant N\\ 0 & \text{其余}n\end{cases}$$

对 $N=5,15,25,100$，求其 DTFT。将它归一化使得 $X(e^{j0})=1$。在区间[$-\pi,\pi$]内绘出归一化的 DTFT 图，评论其结果与 N 的关系.

3.10 对下列每个系统, 求出其频率特性 $H(e^{j\omega})$ 并画出其幅度和相位曲线。

(a) $y(n)=\sum_{m=0}^{6}x(n-m)$

(b) $y(n)=x(n)+2x(n-1)+x(n-2)-0.5y(n-1)-0.25y(n-2)$

(c) $y(n)=2x(n)+x(n-1)-0.25y(n-1)+0.25y(n-2)$

(d) $y(n)=x(n)+x(n-2)-0.81y(n-2)$

(e) $y(n)=x(n)-\sum_{m=0}^{5}(0.6)^m y(n-m)$

3.11 用概念判断，下面的序列中哪些具有纯实数的 DTFT? 哪些具有纯虚数的 DTFT?

(a) $x_1(n)=\begin{cases}|n|, & -N\leqslant n\leqslant N\\ 0, & \text{其它}\end{cases}$　　(b) $x_2(n)=\begin{cases}n^3, & -N\leqslant n\leqslant N\\ 0, & \text{其他}\end{cases}$

(c) $x_3(n)=\begin{cases}0, & 对偶数n \\ 2/\pi n, & 对奇数n\end{cases}$　　(d) $x_4(n)=\begin{cases}0, & n=0 \\ \cos\pi n/n, & |n|>0n\end{cases}$

(e) $x_5(n)=\sin(\omega_c n)/\pi n$

3.12 用概念判断，下面的序列中哪些具有偶序列的 IDTFT？哪些具有奇序列的 IDTFT？

(a) $Y_1(e^{j\omega})=\begin{cases}|\omega|, & 0\leqslant|\omega|\leqslant\omega_c \\ 0, & \omega_c<|\omega|\leqslant\pi\end{cases}$　　(b) $Y_2(e^{j\omega})=\begin{cases}j, & -\pi<\omega<0 \\ -j, & 0<\omega<\pi\end{cases}$

(c) $Y_3(e^{j\omega})=j\omega,\ 0\leqslant|\omega|\leqslant\pi$

3.13 用 DFS 定义求出下列周期序列的 DFS 系数，并用 MATLAB 验证。

(a) $\tilde{x}_1(n)=[2,0,2,0], N=4$　　(b) $\tilde{x}_2(n)=[0,0,2,0,0], N=5$

(c) $\tilde{x}_3(n)=[3,-3,3,-3], N=4$　　(d) $\tilde{x}_4(n)=[j,j,-j,-j], N=4$

(e) $\tilde{x}_5(n)=[1,j,j,1], N=4$

3.14 根据下列周期 DFS 系数,求出周期时间序列 $\tilde{x}(n)$，首先用 IDFS 定义计算，然后用 MATLAB 验证。

(a) $\tilde{X}_1(n)=[5,-2j,3,2j], N=4$　　(b) $\tilde{X}_2(n)=[4,-5,3,-5], N=4$

(c) $\tilde{X}_3(n)=[1,2,3,4,5], N=5$　　(d) $\tilde{X}_4(n)=[0,0,2,0], N=4$

(e) $\tilde{X}_5(n)=[0,j,-2j,-j], N=4$

3.15 12 点序列 $x(n)$为：　$x(n)$={ -3, -2, 3, 4, 2, -1, -1, 2, 4, 3, -2, -3 }

(a) 求出 $x(n)$的 DFT $X(k)$，画出它的幅度和相位曲线（使用 stem 函数）。

(b) 用 MATLAB 画出 $x(n)$的 DTFT $X(e^{j\omega})$ 的幅度和相位曲线。

(c) 验证(a)项中的 DFT 是 $X(e^{j\omega})$ 的采样。采用 hold 函数把两图放在一幅图里。

(d) 有无可能从 DFT $X(k)$重构 DTFT $X(e^{j\omega})$？如果可能，给出重构所需要的内插公式；若不可能，说明不能重构的理由。

3.16 以 DFT 作计算工具，画出下列序列的 DTFT 幅度特性。猜想合理的长度 N，使所做的图有意义。

(a) $x_1(n)=2\cos(0.2\pi n)[\mu(n)-\mu(n-10)]$

(b) $x_2(n)=\sin(0.45\pi n)+\sin(0.55\pi n),\ 0\leqslant n\leqslant 50$

(c) $x_3(n)=3\cdot(2)^n,\ 0\leqslant n\leqslant 50$

(d) $x_4(n)=(-0.5)^n,\ -10\leqslant n\leqslant 10$

(e) $x_5(n)=5\cdot(0.95e^{j\pi/4})^n\mu(n)$

3.17 设 $X(k)=\text{DFT}[x(n)]$，试证明 DFT 的初值定理，即证明 $x(0)=\dfrac{1}{N}\sum\limits_{k=0}^{N-1}X(k)$。

3.18 设 $X(k)$=DFT $[x(n)]$，$Y(k)$=DFT $[y(n)]$，若 $Y(k)=X((k+m)_N)R_N(k)$，证明

$$y(n)=\text{IDFT}[Y(k)]=W_N^{mn}x(n)$$

3.19 设 $X(k)=X_1{*}(k)X_2(k)$，试证明 DFT 的离散相关定理，即证明

$$x(n)=\text{IDFT}[X(k)]=\sum_{m=0}^{N-1}x_1^*(m)\cdot x_2((m+n)_N)\cdot R_N(n)。$$

3.20 对采样周期为 T=0.5 的以下各序列：

(a) $x=[1,\underline{0},-1]$ (b) $x=[\underline{1},-2,1]$ (c) $x=[\underline{-1},2,2,-1]$

（1）用手工草画出它们的幅频谱和相频谱；

（2）求出它们的 DFT 和频点位置；

（3）分析两个结果的关系。

3.21 对上题中各个序列，都在其尾部补两个零，求其频谱并与上题比较。

3.22 用 1024 点 DFT 计算上题各序列的频谱，并与原题的结果做比较。

3.23 实序列 $x(n)$的八点 DFT 的前五个值为 {0.25, 0.125-j0.3, 0, 0.125-j0.06, 0.5}，利用 DFT 性质求出下列各个序列的 DFT。

(a) $x_1(n)=x((2-n)_8)$ (b) $x_2(n)=x((n+5)_{10})$ (c) $x_3(n)=x^2(n)$

(d) $x_4(n)=x(n)⑧\ x((-n)_8)$ (e) $x_5(n)=x(n)e^{j\pi n/4}$

3.24 计算下列序列的 N 点循环卷积 $z(n)$。

(a) $x_1(n)=[1,1,1,1]$, $x_2(n)=\cos(\pi n/4)R_N(n)$; $N=8$

(b) $x_1(n)=\cos(2\pi n/N)R_N(n)$, $x_2(n)=\sin(2\pi n/N)R_N(n)$; $N=32$

(c) $x_1(n)=(0.8)^n R_N(n)$; $x_2(n)=(-0.8)^n\ R_N(n)$; $N=20$

(d) $x_1(n)=nR_N(n)$; $x_2(n)=(N-n)R_N(n)$; $N=10$

(e) $x_1(n)=[1,-1,1,-1]$, $x_2(n)=[1, 0, -1, 0]$; $N=4$

3.25 对于上题所给序列，计算：

(a)$x_1(n)$和 $x_2(n)$的 N+1 点循环卷积 (b)线性卷积 $y(n)=x_1(n)\otimes x_2(n)$

(c)误差序列 $e(n)=z(n)-y(n)$

3.26 采用频域方法，开发一个 MATLAB 函数实现两序列间循环卷积的运算，其格式应为：function x3 = circonvf(x1, x2, N)

```
% 频域循环卷积
% x3 = circonvf(x1,x2,N)
% x3 = 长度为 N 的卷积结果
% x1 = 长度 <= N 的输入序列
% x2 =  长度 <= N 的输入序列
%  N = 循环缓冲器的长度
```

3.27 (1) 计算下面各序列的 DFT，利用 stem 函数画出 DFT 序列的实部。

(i) $x_1(n)=3\cos(0.04\pi n)R_{200}(n)$;

(ii) $x_2(n)=[4\cos(0.1\pi n)-3\cos(1.9\pi n)]R_{20}(n)$;

(iii) $x_3(n)=[1+2\cos(0.5\pi n)+\cos(\pi n)]R_{100}(n)$;

(iv) $x_4(n)=\cos(25\pi n/16)R_{64}(n)$.

(2) 若将上述的各截取方波宽度都增加或减小 1，得出的结果如何？为什么？

(3) 对于 $x(n)=A\cos(2\pi mn/N)R_N(n)$,其中 m 为整数。注意 $x(n)$在 N 个样本中包含余弦的 m 个周期，这是一个由窗口截取的无泄漏的余弦序列。证明：它的 DFT 为实序列。

$$X(k)=\frac{AN}{2}\delta(k-m)+\frac{AN}{2}\delta(k-N+m) \qquad 0\leqslant k\leqslant N-1,\ 0<m<N$$

3.28 设 $x(n)=A\cos(\omega_0 n)R_N(n)$，其中 ω_0 为实数。

(a)利用 DFT 性质，证明 $X(k)$的实部和虚部为

$X(k)=X_R(k)+jX_I(k)$ $X(k)=X_R(k)+jX_I(k)$。

其中：

$$X_R(k)=\frac{A}{2}\cos\left[\frac{\pi(N-1)}{N}(k-f_0N)\right]\frac{\sin[\pi(k-f_0N)]}{\sin[\pi(k-f_0N)/N]}+$$

$$\frac{A}{2}\cos\left[\frac{\pi(N-1)}{N}(k+f_0N)\right]\frac{\sin[\pi(k-N+f_0N)]}{\sin[\pi(k-N+f_0N)/N]}$$

$$X_I(k)=-\frac{A}{2}\sin\left[\frac{\pi(N-1)}{N}(k-f_0N)\right]\frac{\sin[\pi(k-f_0N)]}{\sin[\pi(k-f_0N)/N]}-$$

$$\frac{A}{2}\sin\left[\frac{\pi(N-1)}{N}(k+f_0N)\right]\frac{\sin[\pi(k-N+f_0N)]}{\sin[\pi(k-N+f_0N)/N]}$$

(b) 上述结果说明，余弦信号的初始频率 ω_0 泄漏进其他频率里，形成时间有限序列的谐波，因此把它叫做余弦泄漏性质。它是由于在非整数周期上对带宽有限的周期余弦信号进行采样产生的必然结果。用 $x(n)$的周期延拓 $\tilde{x}(n)$ 的特性解释此结果。

(c) 用 $x(n)=\cos(5\pi n/99)R_{200}(n)$ 验证泄漏特性，利用 stem 函数画出 DFT 序列的实部和虚部。

3.29 试证明循环卷积符合交换律、结合律和分配律。

3.30 设 x=[-1:0.5:3], h=[3:6];

(a) 列出用向量-矩阵乘法求两者的线性卷积的表示式，写出构成其线性卷积矩阵的 MATLAB 语句。

(b) 列出用向量-矩阵乘法求两者的循环卷积的表示式，写出构成其循环卷积矩阵的 MATLAB 语句。

3.31 设 $G(k)$和 $H(k)$分别为两个七点序列 $g(n)$和 $h(n)$的七点 DFT。

(a) 若 $G(k)$=[1+j2, -2+j3, -1 -j2, 0, 8+j4, -3+j, 2+j5]而 $h(n)=g((n-3)_7)$，不作 DFT，直接求 $H(k)$。

(b) 若 $g(n)$=[-3.1, 2.4, 4.5, -6, 1, -3, 7]而 $H(k)=G((k-4)_7)$，不作 DFT 和 IDFT，直接求 $h(n)$。

第 4 章　信号频谱的高效计算

4.1　各种傅里叶变换及其相互关系

4.1.1　傅里叶变换的主要类型

总结连续信号和离散信号的傅里叶变换（级数）在频域的特点，可以归纳成表 4.1.1。其中连续信号的傅里叶变换（级数）是在信号与系统课程中推导的。可以看出它们之间的如下对偶关系。

表 4.1.1　各类傅里叶变换中信号和频谱的特征

变换名称	时域信号（傅里叶反变换）	频谱曲线（傅里叶变换）
(连续)傅里叶变换(CFT)	连续信号	连续频谱
(连续)傅里叶级数(CFS)	周期性，连续信号	离散频谱
离散时间傅里叶变换(DTFT)	离散信号	周期性，连续频谱
离散傅里叶级数(DFS)	周期性，离散信号	周期性，离散频谱
离散傅里叶变换(DFT)	有限长离散信号(隐含周期)	有限长离散频谱(隐含周期)

（1）时域的周期性对应于频域的离散化。即周期信号的频谱是集中在若干等距离分割的频点上的脉冲函数，它们代表各个谐波分量，所以具有倍频关系。

（2）时域的离散化对应于频域的周期性。即离散信号的频谱是连续的并具有周期性。其主周期就是奈奎斯特频率范围$[-\pi/T, \pi/T]$。

（3）根据上面两条，可知周期性的离散时域信号将对应于离散并具有周期性的频谱，也就是离散傅里叶级数(DFS)。正反变换都离散有利于数值计算，但周期信号延伸到无穷又不利于数值计算。

（4）把时域和频域数据长度都限于主周期，并且使其长度相等，形成了一种标准的对称的离散傅里叶变换(DFT)。它既离散，又长度有限，适合于计算机数值计算。对这样的傅里叶变换，人们开发了可以高效地进行计算的方法，称为快速傅里叶变换(FFT)。对于其他任意形式的信号，人们也尽量对它们进行修改，以便利用 FFT 来解决其频谱计算问题。这就带来怎样修改，误差才小，以及应该怎样修改等问题。

本章主要从工程计算的角度出发，先介绍了 FFT 算法，然后探讨如何快速而准确地利用 FFT 计算任意信号的频谱问题。下面先把几种变换对偶的公式列出，以便引用。

这里要说明一下所用的符号规则。因为本书着重讨论数字信号，所以离散序列和它的频谱及傅里叶系数 a_k 都不加下标，模拟信号和它的频谱及傅里叶系数 a_k 都加下标 a。连续频谱和离散频谱主符号相同，用自变量下标来区分。连续频谱表为 $X(\omega)$或 $X(\Omega)$；离散频谱表为 $X(\omega_k)$, $X(\Omega_k)$ 或 $X(k)$。

（1）连续周期信号的傅里叶级数(CFS)

周期连续信号的傅里叶级数对

$$a_{ka}=\frac{1}{T}\int_{-T/2}^{T/2}x_a(t)e^{-jk\Omega_0 t}dt \qquad k=0,\ \pm1,\ \pm2,\ \cdots \tag{4.1.1a}$$

$$x_a(t)=\sum_{k=-\infty}^{\infty}a_{ka}e^{jk\Omega_0 t} \qquad a_{ka}=|a_{ka}|e^{j\varphi_k} \tag{4.1.1b}$$

其中 T 为信号周期，$\Omega_0=2\pi/T$ 信号的基波频率。不难看出，它的频谱分量为离散值，而时域则是连续量。因此正变换为积分运算，而反变换为求和运算，即原信号为各次谐波之和。

（2）连续非周期信号的傅里叶变换(FT 或 CFT)

非周期连续信号的傅里叶变换对可以由(4.1.1)式中把 T 无限加大或 $\Omega_0=2\pi/T$ 无限减小而得到：

$$X_a(j\Omega)=\int_{-\infty}^{\infty}x_a(t)e^{-j\Omega t}dt \tag{4.1.2 a}$$

$$x_a(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}X_a(j\Omega)e^{j\Omega t}d\omega \tag{4.1.2 b}$$

（3）离散时间傅里叶变换（DTFT）

非周期性离散信号的傅里叶变换（其频率取连续量 Ω）

$$X(j\Omega)=F[x(n)]=\sum_{-\infty}^{\infty}x(nT)e^{-j\Omega nT} \quad \Omega=[\Omega_1,\Omega_2,\cdots]\in\left[-\pi/T,\pi/T\right) \tag{4.1.3a}$$

用数字频率 $\Omega=\omega/T$ 时

$$X(j\omega)=F[x(n)]=\sum_{-\infty}^{\infty}x(n)e^{-j\omega n} \qquad \omega=[\omega_1,\omega_2,\cdots]\in[-\pi,\pi) \tag{4.1.3b}$$

$$x(n)=\frac{T}{2\pi}\int_{\pi/T}^{\pi/T}X(j\Omega)e^{j\Omega nT}d\Omega=\frac{1}{2\pi}\int_{-\pi}^{\pi}X(j\omega)e^{j\omega n}d\omega \tag{4.1.3c}$$

此时时域信号是离散的，而它们的傅里叶变换 DTFT 则是连续的。所以正变换是用求和式，而反变换则是用积分式。由于 DTFT 的周期性，积分范围取为奈奎斯特频率区间。

（4）周期性离散时间信号的傅里叶级数（DFS）

对于无限长周期序列 $\tilde{x}(n)$，它不满足绝对可加条件，因而傅里叶变换无界，但其傅里叶变换与周期数之比却趋向于常数。可以取来表征其频谱，它就是离散的傅里叶级数，即

$$\tilde{X}(k)=\sum_{n=0}^{N-1}\tilde{x}[n]e^{-j\frac{2\pi}{N}kn}=N\cdot a_k \qquad -\infty<k<\infty \tag{4.1.4 a}$$

$$\tilde{x}(n)=\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)e^{j\frac{2\pi}{N}kn}=\sum_{k=0}^{N-1}a_k e^{j\frac{2\pi}{N}kn} \quad -\infty<k<\infty \tag{4.1.4 b}$$

可以看出，$\tilde{x}(n)$ 和 $\tilde{X}(k)=N\cdot a_k$ 都是离散周期序列，所以正反变换都用求和运算，很适合计算机数字计算。问题在于这时两者都是无限序列。

（5）离散傅里叶变换（DFT）

考虑 DFS 的主值区间，把频率向量的长度取得与时间样本数 N 相同，这样计算机计算可以有规范化的程序。引入

$$\Omega_k=kD=k\left(\frac{2\pi}{NT}\right) \quad \text{或} \quad \omega_k=T\Omega_k=k\left(\frac{2\pi}{N}\right)$$

其中
$$D=\frac{2\pi}{NT}=\frac{2\pi}{N}F_s \tag{4.1.5}$$

为频率分辨率，又引入旋转算子符号 $W_N=e^{-j\frac{2\pi}{N}}$，则离散傅里叶级数对的关系(4.1.4)式就变成

$$X(k)=F[x(n)]=\sum_{n=0}^{N-1}x(n)W_N^{nk} \qquad k=0,1,\cdots,N-1 \tag{4.1.6a}$$

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-nk} \qquad n=0,1,\cdots,N-1 \tag{4.1.6b}$$

公式(4.1.6 a)和 (4.1.6 b)是计算 DFT 和 IDFT 的规范形式。

4.1.2 DFT 与其他变换的直接关系

在上述的各种傅里叶变换中，能够用计算机以很高效率计算的，只有离散傅里叶变换 DFT。要高效地计算任意信号的频谱，必须弄清各种变换与 DFT 的关系。弄清把 DFT 用在各种形式的信号上有什么问题？会带来哪些误差？如何估计这些误差的数量？以及把这些误差减少到允许值以下的措施。总之速度和精度是用数值计算解决问题时要面对的永恒主题，本章将重点讨论这个问题，并用它来解决频谱的分析计算。为此，先把 DFT 和其他几种傅里叶变换的关系简要归纳一下。

（1）DFT 与 DFS 的关系

在概念上，DFS 是对应于无限长周期性的时间序列，而 DFT 则只考虑的其主值周期内的信号。在频率域，DFS 也是离散取值并周期性和无限延伸的，而 DFT 只对应于它的主值区间。所以两者之间在主值区间是完全的一一对应的精确关系。

$$a_k=\frac{X(k)}{N} \quad 或 \quad X(k)=a_k\cdot N \quad k=0,1,\cdots,N-1 \tag{4.1.7}$$

（2）DFT 与 DTFT 的关系

在时域，两者都是离散时间序列 $x(n)$，在频域，DTFT 是连续周期曲线 $X(\omega)$，而 DFT 则是与时间序列同长的离散频谱序列 $X(k)$。DFT 是 DTFT 在主值区间的等间隔采样，DTFT 则是 DFT 在密集的频点上的插值，并作周期延拓。插值当然就有精度问题。所以由 DTFT 求 DFT 是精确的；而由 DFT 求 DTFT 就得考虑截断引起的精度问题。

对 $X(\omega)$采样求（$-\pi/T,-\pi/T$）区域内 N 个等间隔频点的 $X(k)$的关系：

$$X(k)=X\left(k\frac{2\pi}{NT}\right)=X(\omega)\Big|_{\omega=k\frac{2\pi}{NT}} \tag{4.1.8}$$

由对 $X(k)$插值及周期延拓求全频段的 DTFT $X(\omega)$ 的关系如下（可见(3.5.9)式）。

$$X(\omega)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)\frac{\sin[(\omega N-2\pi k)/2]}{\sin[(\omega N-2\pi k)/2N]}\cdot e^{-j[\omega-(2\pi k/N)]\cdot[(N-1)/2]} \tag{4.1.9}$$

（3）DTFT 与 CTFT 的关系

连续信号的 CTFT 即频谱 $X_a(\Omega)$ 与把它采样后的离散傅里叶变换(DFT) $X(k)$ 之间没有直接的变换关系。要建立两者的关系，先把连续信号采样，求其 DTFT，得到离散序列 $x(n)$ 的连续频谱 $X(\omega)$；然后要把 $X(\omega)$ 与 $X_a(\Omega)$ 这两个连续频谱之间的数学关系建立起来，本章第 4.4 节将讨论并解决这个问题。它就是著名的采样定理，也称为奈奎斯特定理。其结果是

$$X(\omega)=\frac{1}{T}\sum_{k=-\infty}^{\infty}X_{a}\left(\Omega-k\frac{2\pi}{T}\right)\qquad \Omega=\frac{\omega}{T}\tag{4.1.10}$$

所以，按(4.1.10)式知道连续信号频谱 $X_a(\Omega)=X_a(\frac{\omega}{T})$ 求数字频谱 $X(\omega)$ 是没问题的，求出 $X(\omega)$ 后，用(4.1.10)式即可求出 $X(k)$。

用(4.1.10)式反求，即由 $X(\omega)$ 求 $X_a(\Omega)$ 则不一定求得出。只有连续信号频带 Ωc 较窄而采样周期 T 足够小，满足奈奎斯特条件$|\Omega cT|<\pi$，因而频率泄漏可以忽略不计时，才可以得到由 $X(\omega)$ 求 $X_a(\Omega)$ 的下列近似解。

$$X_a(\frac{\omega}{T})=FT[x_a(t)]\approx\begin{cases}T\cdot X(\omega) & |\omega|\leqslant\pi\\ 0 & |\omega|\leqslant\pi\end{cases}\tag{4.1.11}$$

所以反求的过程是：第一步，由 $X(k)$ 求 $X(\omega)$，这里要考虑内插误差；第二步，由 $X(\omega)$ 求 $X_a(\Omega)$，这里要考虑泄漏误差。

（4）DFT 与 CTFS

连续时间周期信号的傅里叶级数(CTFS) a_{ka} （k=0, 1, … ∞）是一个无限长离散序列，而 DFT 无论在时域和频域都是有限长并且离散的。由 CTFS 求 DFT 只是取出其一部分，没有误差问题。而由 DFT 求 CTFS 要先经过由 DFT 求 DTFS，再经过由 DTFS 求 CTFS 的两步。如前所述，前一步问题不大，而后一步也要求采样序列的 DTFS 确实代表了 CTFS 的全部或主要信息，也即要求信号的频谱足够窄，因而频率泄漏可以忽略不计。此时可由下列关系由 $X(k)$ 求得 a_{ka}。

$$a_{ka}\approx a_k=\frac{X(k)}{N},\qquad -\frac{N}{2}+1<k\leqslant\frac{N}{2}$$

各个逆变换之间的关系可依次类推得到。本章在讲到相关的部分时会加以说明，不在此处罗列。

4.2　快速傅里叶变换(FFT)

前面介绍的 DFT 是时域和频域信号均为离散的惟一变换，它适用于有限时间序列。尽管它是可数值计算的，但如果直接按定义（4.1.6）式来实现，当序列长度很大时，运算时间很长，占用内存空间很大。1965 年，Cooly 与 Tukey 提出了大幅度减少 DFT 中的计算量的方法，这使得 DFT 真正得到应用并引起了数字信号处理在广泛的领域内得到发展。同时，它也引起其他高效率算法的探讨。所有这些高效率算法统称为快速傅里叶变换。

4.2.1　时域抽取(DIT)基二 FFT 算法

考虑一个复数序列 $x(n), n=0,1,\cdots,N-1$，它是长度为 N 的有限序列，它的离散傅里叶变换是：

$$X(k)=\sum_{n=0}^{N-1}x[n]\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}kn}=\sum_{n=0}^{N-1}x[n]W_N^{kn}\qquad 0\leqslant k\leqslant N-1\tag{4.2.1}$$

其中

$$W_N=\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}}\tag{4.2.2}$$

直接计算（4.2.1）式对每个 k 需要 N 次复数乘法和（N−1）次复数加法，因此对所有的 k 完成(4.2.1)式的计算需要 N^2 次复数乘法和 $N(N-1)$次复数加法。当 N 增加时，运算次数将按 N^2 的规律快速增长，因此有必要寻求一个更为高效的方法来计算 DFT。下面将讨论其中的一个方法。

设 N 可以被 2 整除，可以把 $x(n)$分成两个子序列 $x_1(n)$和 $x_2(n)$如下：

$$\begin{aligned}x_1(n)&=x(n)=[x(0),x(2),\cdots,x(N-2)]\\x_2(n)&=x(2n+1)=[x(1),x(3),\cdots,(N-1)]\end{aligned}\tag{4.2.3}$$

所以 $x_1(n)$，$x_2(n)$中分别包括了 $x(n)$中的偶数项和奇数项，显然每个序列的长度 $N_1=N/2$，所以它们的 DFT 为

$$X_1(m)=\sum_{n=0}^{N_1-1}x_1(n)W_{N_1}{}^{mn}\tag{4.2.4}$$

$$X_2(m)=\sum_{n=0}^{N_1-1}x_2(n)W_{N_1}{}^{mn}\tag{4.2.5}$$

其中

$$W_{N_1}=\mathrm{e}^{-\mathrm{j}2\pi/N_1}\tag{4.2.6}$$

而 $m=0,1,\cdots,N_1-1$，注意，$X_1(m)$ 和 $X_2(m)$ 具有周期为 $N_1=N/2$ 的隐含周期性：

$$X_i(m)=X_i(m+rN_1)\tag{4.2.7}$$

对所有的整数 r 和 i=1, 2 成立。然后用 $X_1(m)$ 和 $X_2(m)$ 来表示 $X(m)$，利用

$$W_{N_1}=\mathrm{e}^{-\mathrm{j}2\pi/N_1}=\mathrm{e}^{-\mathrm{j}2\pi/(N/2)}=W_N^2$$

重写(4.2.1)式如下：

$$\begin{aligned}X(m)&=X(0)W_N^0+X(2)W_N^{2m}+\cdots+x(N-2)W_N^{(N-2)\,m}+\\&\quad X(1)W_N^m+X(3)W_N^{3m}+\cdots+x(N-1)W_N^{(N-1)m}\\&=x_1(0)W_{N_1}^0+x_1(1)W_{N_1}^m+\cdots+x_1(N_1-1)W_{N_1}^{(N_1-1)m}+\\&\quad W_N^m[x_2(0)W_{N_1}^0+x_2(1)W_{N_1}^m+\cdots+x_2(N_1-1)W_{N_1}^{(N_1-1)m}\end{aligned}\tag{4.2.8}$$

因此可以得到

$$X(m)=X_1(m)+W_N^mX_2(m)\qquad m=0,1,\cdots,N-1\tag{4.2.9}$$

现在计算(4.2.9)式需要的乘法次数，（4.2.4）式和（4.2.5）式中每一个 $X_i(m)$需要 N_1^2 次复数乘法，此外，用(4.2.9)式中求出 $X(m)$ 还需要 N 次复数乘法。因此，总计需要的乘法次数为 $\left(\frac{N}{2}\right)^2+\left(\frac{N}{2}\right)^2+N=N+\frac{N^2}{2}$，而需要的加法次数为 $\frac{N}{2}\left(\frac{N}{2}-1\right)+\frac{N}{2}\left(\frac{N}{2}-1\right)+N=\frac{N^2}{2}$。

与直接计算（4.2.1）式所需的 N^2 次乘法和 N（N−1）次加法相比，差不多减少了一半。如果把 $x_1(n)$和 $x_2(n)$再分解成两个子序列，那么，总的运算次数又能减少一半，这就是快速计算傅里叶变换的一种思路。

现在详细地把这个算法在 $N=8$ 的过程讨论清楚，这时公式(4.2.1)、(4.2.4)、(4.2.5)成为

$$X(m)=x(0)+x(1)W_8^m+x(2)W_8^{2m}+x(3)W_8^{3m}+x(4)W_8^{4m}+x(5)W_8^{5m}+x(6)W_8^{6m}+x(7)W_8^{7m}$$
$$m=0,1,2,3,\cdots$$

$$X_1(m)=x(0)+x(2)W_8^{2m}+x(4)W_8^{4m}+x(6)W_8^{6m}\qquad m=0,1,2,3,\cdots$$

$$X_2(m)=x(1)W_8^m+x(3)W_8^{3m}+x(5)W_8^{5m}+x(7)W_8^{7m} \qquad m=0,1,2,3,\cdots$$

其中$W_N=W_8=\mathrm{e}^{-\mathrm{j}\pi/4}$。注意 $X(m)$的周期为 8 而 $X_1(m)$的周期为 4，进而再把 $x_1(n)$分配成为两个序列[x(0), x(4)]和[x(2), x(6)]，它们的长度为 N_2=2。定义

$$W_{N2}=W_2^1=W_8^4=\mathrm{e}^{-\mathrm{j}\pi}$$

得出的 DFT 是

$$X_{11}(m)=x(0)W_{N_2}^0+x(4)W_{N_2}^m=x(0)+x(4)W_8^{4m} \qquad m=0,1$$

$$X_{12}(m)=x(0)W_{N2}^0+x(4)W_{N2}^m=x(0)+x(4)W_8^{4m} \qquad m=0,1$$

它们的周期为 2，将其代入 $X_1(m)$中，得到

$$X_1(m)=X_{11}(m)+W_8^{2m}X_{12}(m) \qquad m=0,1,2,3$$

同样，$X_2(m)$也可由长度为 2 的 DFT 算得。这个计算 DFT 的过程在图 4.2.1 中画出，可以看出，其中每一项都是从(4.2.9 a)形式的公式算出的。

可以把图 4.2.1 中的过程对 $N=2^k$ 的情况加以普遍化。第一步是要把输入数据 $x(n)$ 的位置向量适当的排序，排序方法将在后面讨论。然后把 DFT 分成 $K=\log_2N$ 段来计算，在每一段要计算 N 个数，每个数由一个(4.2.9 a)式算出，它需要一次复数运算，这样每一段需要 N 次复数乘法。现在有 $\log_2N$ 段，因此，整个 FFT 大概总共需要 $N\times\log_2N$ 次复数乘法。

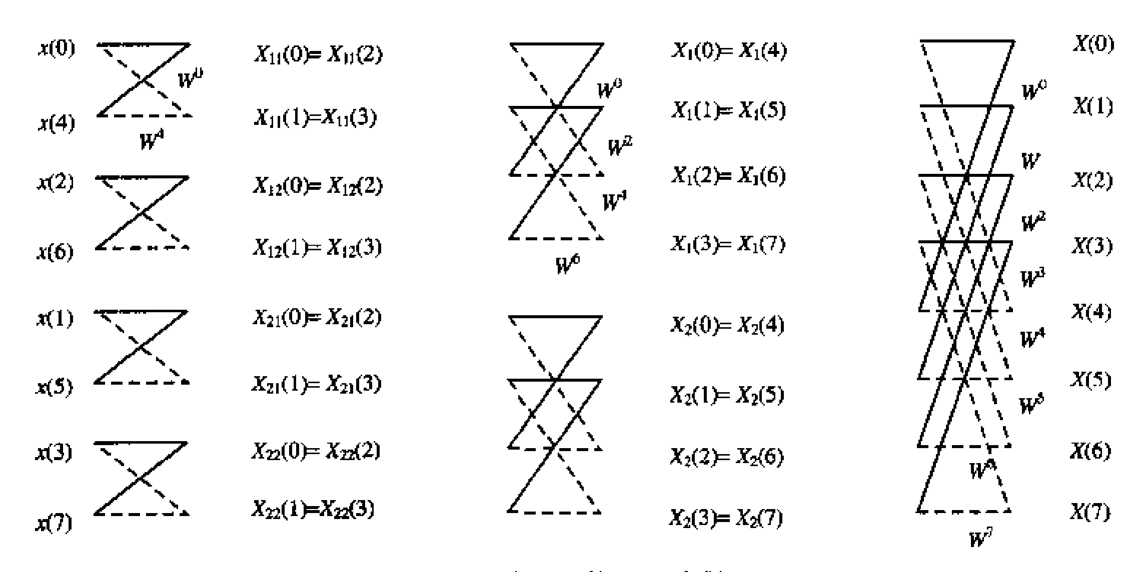


图 4.2.1　对 N=8 的 FFT 流程

表 4.2.1　DFT 和 FFT 复数运算次数的比较

运算次数 / 序列 N 长度	DFT		FFT	
	乘（×）	加（+）	乘（×）	加（+）
2	4	2	1	2
8	64	56	12	24
32	1024	922	80	160
64	4096	4022	192	384
1024	1048576	1048576	5120	10240
2^{20}	$\approx10^{12}$	$\approx10^{12}$	$\approx10^7$	$\approx2\times10^7$

此外因为$W_N^0=1$, $W_N^{N/2}=-1$，用(4.2.9 a)式的计算还有潜力可挖。比如计算它在单位圆上对面的点$X(m+N/2)$。考虑到

$$W_N^{(m+N/2)}=W_N^{N/2}W_N^m=-W_N^m$$

$$X(m+N/2)=X_1(m)-W_N^mX_2(m)\qquad m=0,1,\cdots,N/2-1\qquad(4.2.10)$$

把公式（4.2.9）和（4.2.10）两式结合起来的运算结构图如图 4.2.2 所示。根据这种运算结构的形状，称之为蝶形运算。因为复数乘法项$W_N^mX_2(m)$可以公用，比单纯用（4.2.9）式的乘法次数减少一半。从图 4.2.1 和图 4.2.3 可以看到，FFT 的运算最后都可以用若干个蝶形运算来实现。所以大概有一半的乘法还可以节省，所以整个 FFT 大体上只需要 $0.5N\times\log_2 N$ 次复数乘法和 $N\times\log_2 N$ 次复数加法。表 4.2.1 列出了直接计算 DFT 和 FFT 所需的复数乘法和复数加法的数目。对于 N 很大的情况，FFT 所节省的运算次数是很显著的。例如，对于 $N=2^{10}=1024$ 来说，乘法的次数减到了 1/200 而加法的次数减到了 1/100。这意味着原来可能需要 2～3min 的 DFT 计算程序，采用 FFT 后只需要 1s 就能完成。

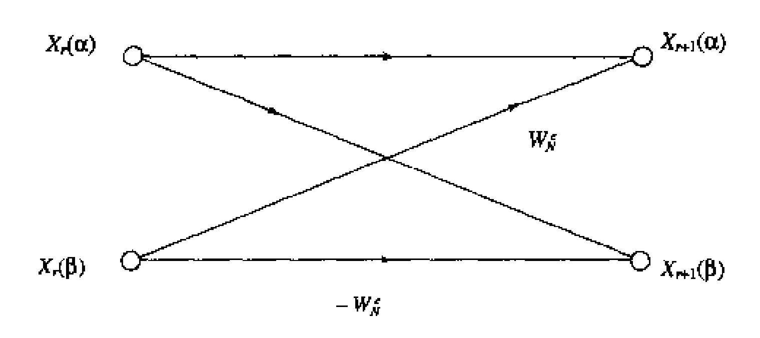


图 4.2.2　蝶形运算的结构图

再考虑节省存储量的问题。按图 4.2.1，输入数据排列在最左边的一列上，共需要 N 个存储单元。以左上角的两项为例，一旦算出了$[X_{11}(0),X_{11}(1)]$，以后的计算就不再用到$[X(0),X(4)]$了，因此就可以把$[X_{11}(0),X_{11}(1)]$存在$[x(0),x(4)]$的存储单元中。对其他输入数据计算对也可以作同样的安排，这样，计算时就只需要输入数据的 N 个存储单元，计算完成后，其中就存储了输出结果，不需要任何额外的存储单元。这称为原位计算。原位计算把原来需要 $N\times N$ 个单元减少到 N 个单元，所以 FFT 在时间和空间上都大大节省了计算资源。

再来讨论图 4.2.1 上输入数据的排列问题。它是按 FFT 过程中二进制位倒序的规则得到的。首先把 n 按二进制排列，使$x(n)=x[k_2k_1k_0]$,其中k_i只取 0 或 1，如图 4.2.3 所示。然后把二进制的位左右倒序成为$[k_0k_1k_2]$，再把倒序后的下标恢复成自然数，就构成了如图 4.2.1 所示的输入序列。从该图中 FFT 算法构成的方法可以验证倒序的正确性。

假如把$x(n)$写成$x[k_2k_1k_0]$，则把$x[k_2k_1k_0]$分解为两个序列$x_1(n)$和$x_2(n)$时，就可根据$k_0=0$或 1 来区别。把$x_1(n)$和$x_2(n)$再分解为子序列，又可根据k_1来区分。这个分解的过程可以用图 4.2.4 来表示。可以看到，在这个过程中，二进制下标$[k_2k_1k_0]$被倒序了，这也证明了图 4.2.1 中输入排列次序的正确性。

二进制编码		位倒序		重新排列		解码	
$x(0)$	→	$x(000)$	→	$x(000)$	→	$x(000)$	→ $x(0)$
$x(1)$	→	$x(001)$	→	$x(100)$		$x(001)$	→ $x(4)$
$x(2)$	→	$x(010)$	→	$x(010)$		$x(010)$	→ $x(2)$
$x(3)$	→	$x(011)$	→	$x(110)$		$x(011)$	→ $x(6)$
$x(4)$	→	$x(100)$	→	$x(001)$		$x(100)$	→ $x(1)$
$x(5)$	→	$x(101)$	→	$x(101)$		$x(101)$	→ $x(5)$
$x(6)$	→	$x(110)$	→	$x(011)$		$x(110)$	→ $x(3)$
$x(7)$	→	$x(111)$	→	$x(111)$	→	$x(111)$	→ $x(7)$

图 4.2.3　位倒序的处理过程

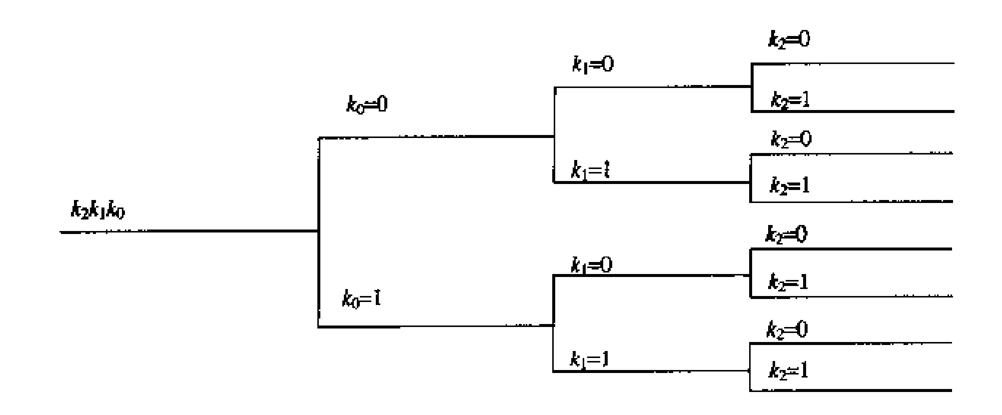


图 4.2.4　位倒序正确性的验证

4.2.2　用 MATLAB 程序描述 FFT 算法

为了能用简明的语言总结上述基二 DIT FFT 算法，下面用 MATLAB 对它进行描述。不仅因为 MATLAB 的简洁，而且还可以通过运行来检验程序的正确性和速度。程序写成函数形式，命名为 myditfft。编程的目的是达到思路清楚，并未考虑其运行的效率。因为用 MATLAB 这样的高级语言，优点就是程序好懂。它是以解释方式工作的，本身就快不了。在跑不快的平台上花太多心思去提高速度，是不值得的。程序清单如下：

```
function y=myditfft(x)
% 本程序对输入序列 x 实现 DIT-FFT 基 2 算法，点数取大于等于 x 长度的 2 的幂次
% ------------------------------------------------------------
% y=myditfft(x)
%
m=nextpow2(x);N=2^m;                % 求 x 的长度对应的 2 的最低幂次 m
if length(x)<N
   x=[x,zeros(1,N-length(x))];      % 若 x 的长度不是 2 的幂，补零到 2 的整数幂
end
```

```
nxd=bin2dec(fliplr(dec2bin([1:N]-1,m)))+1;   % 求1:2^m数列的倒序
y=x(nxd);                                    % 将x倒序排列作为y的初始值
for mm=1:m                                   % 将DFT作m次基2分解,从左到右, 对每次分解作DFT运算
Nmr=2^mm;u=1;                                % 旋转因子u初始化为WN^0=1
WN=exp(-i*2*pi/Nmr);                         % 本次分解的基本DFT因子WN=exp(-i*2*pi/Nmr)
  for j=1:Nmr/2                              % 本次跨越间隔内的各次蝶形运算
    for k=j:Nmr:N                            % 本次蝶形运算的跨越间隔为Nmr=2^mm
      kp=k+Nmr/2;                            % 确定蝶形运算的对应单元下标
      t=y(kp)*u;                             % 蝶形运算的乘积项
      y(kp)=y(k)-t;                          % 蝶形运算的加法项
      y(k)=y(k)+t;                           % 蝶形运算的加法项
    end
    u=u*WN;                                  % 修改旋转因子,多乘一个基本DFT因子WN
  end
end
```

对这个程序中的某些语句需要做一些说明。

（1）y=nextpow2(x)：如果 x 是一个数，用来求最靠近 x 并大于 x 的二的幂次。例如 nextpow2(9.5)=4,因为最靠近 9.5 并比它大的 2 的乘幂是 2^4=16；如果 x 是一个向量，则 nextpow2(x)=nextpow2(length(x))。即用来求 x 长度的最近的二的幂次。

（2）MATLAB 信号处理工具箱中也有倒序函数 bitrevorder，为了弄清意义，宁可自编。

（3）dec2bin(x)把十进制数 x 转为二进制。bin2dec(x)把二进制数 x 转为十进制，fliplr(x)把 x 数组排列左右反转。

（4）变量 mm 的意义相当于 FFT 分段的序数，mm 最大相当于 $\log_2 N$。

此子程序的正确性经过检验之后，下面来检验它的运行速度。参照的程序有两个：一个是比它快的，即 MATLAB 提供的内部函数 fft。这个程序是用机器的目的码写成的，无法列出可读的清单；另一个是比它慢的，由 DFT 定义直接写出的向量-矩阵乘法的程序。这里不妨再写一遍：

```
function y=mydft(x)
N=length(x); n=0:N-1; k=n;
WN=exp(-j*2*pi/N);
nk=n'*k; WNnk=WN.^nk;
Xk=x*WNnk;y=Xk;
```

为了检验程序运行时间，可编写如下程序：

```
K=input('K= ');                  % 设定数据长度的二的幂次K
x=randn(1,2^K);                  % 先生成一个x向量：生成向量的时间不予计入
tic,X=fft(x),toc                 % 测试fft子程序所需运行时间
tic,X=myditfft(x),toc            % 测试myditfft子程序所需运行时间
tic,X=mydft(x),toc               % 测试mydft子程序所需运行时间
```

其中 tic 和 toc 是启动和停止秒表的命令。当 K=10 时，在作者的计算机上得到的结果是：fft 函数，0.01s(实际上还要小)；myditfft 函数，0.32s；mydft 函数，22s；

4.2.3 基四和其他基的 FFT 算法

上述的 DIT FFT 算法称为分裂-组合算法，它还可以进一步普遍化，设总点数 N 可以分解为两个任意整数的乘积 N=$N1$*$N2$，则就可以把 N 点 FFT 分解为两段，第一段是进行 $N2$ 次 $N1$ 点 DFT，第二段再进行 $N1$ 次 $N2$ 点 DFT，在两段之间，需要把 N 个中间数据都乘以旋转因子 $W_N^{\ k}$。用较小的 DFT 组合成大的 DFT，

这是它的基本思想。按照这个分段的思想，把离散傅里叶变换式

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk} \qquad 0\leqslant k\leqslant N-1 \tag{4.2.11}$$

中的 n 与 k 写为：

$$n=N_1 l+m \qquad 0\leqslant l\leqslant N_2-1,\ 0\leqslant m\leqslant N_1-1$$

$$k=p+N_2 q \qquad 0\leqslant p\leqslant N_2-1,\ 0\leqslant q\leqslant N_1-1$$

改写序列 $x(n)$和 $X(k)$的下标，分别成为数组 $x(1,m)$和 $X(p,q)$，则（4.2.11）式可写成

$$\begin{aligned}X(p,q)&=\sum_{m=0}^{N_1-1}\sum_{l=0}^{N_1-1}x(l,m)W_N^{(N_1 l+m)(p+N_2 q)}\\&=\sum_{m=0}^{N_1-1}\left\{W_N^{mp}\left[\sum_{l=0}^{N_2-1}x(l,m)W_N^{N_2 lp}\right]\right\}W_N^{N_2 mq}\end{aligned} \tag{4.2.12}$$

注意恒等式

$$\begin{aligned}W_N^{N_1}&=\mathrm{e}^{-\mathrm{j}2\pi\cdot N_1/N}=\mathrm{e}^{-\mathrm{j}2\pi/N_2}=W_{N_2}\\W_N^{N_2}&=\mathrm{e}^{-\mathrm{j}2\pi\cdot N_2/N}=\mathrm{e}^{-\mathrm{j}2\pi/N_1}=W_{N_1}\end{aligned} \tag{4.2.13}$$

（4.2.12）式可以化为

$$X(p,q)=\sum_{m=0}^{N_1-1}\left\{W_N^{mp}\left[\sum_{l=0}^{N_2-1}x(l,m)W_{N_2}^{lp}\right]\right\}W_{N_1}^{mq} \tag{4.2.14}$$

因此可按其括号的次序，从内到外，分以下三步实现公式（4.2.14）。

（1）按下标 l 计算 N_2点 DFT，需做乘法 N_2^2 次，对每一个 m 都要算一遍，共需计算 N_2遍，总共需乘法 $N_2^2\times N_1$次。

$$F(p,m)=\sum_{l=0}^{N_2-1}x(l,m)W_{N_2}^{lp} \qquad m=0,1,\cdots,N_1-1 \tag{4.2.15}$$

（2）N 个数据 $F(\mathrm{p},m)$ 都要乘以旋转因子 W_N^{mp}，总共要作 N 次乘法。

$$G(p,m)=W_N^{mp}\cdot F(p,m) \qquad 0\leqslant m\leqslant n_1-1,\ 0\leqslant p\leqslant N_2-1 \tag{4.2.16}$$

（3）按下标 m 计算 N_1点 DFT，需做乘法 N_1^2 次，对每一个 p 都要算一遍，共需计算 N_2遍，总共需乘法 $N_1^2\times N_2$次。

$$X(p,q)=\sum_{m=0}^{N_1-1}G(p,m)W_{N_1}^{mq} \tag{4.2.17}$$

把这三步的乘法次数相加，进行一次分裂-组合算法所用的复数乘法次数为

$$N_1\cdot N_2^2+N+N_2\cdot N_1^2=(N_1+1+N_2)\cdot N \tag{4.2.18}$$

因为有 $N_1\cdot N_2=N$，（$N_1,N_2\neq 1$）$N1+N2\leqslant N$。所以，只要 N 足够大，这种分解总是可以节省运算次数的。等号出现于 $N=4$ 时，即把 $N=4$ 分解为 $N_1=N_2=2$ 的两个 FFT，乘法次数并不节省，考虑到（4.2.18）式右端还多一个 1，反而不合算。推广来说，如果 N 可以分解为 ν 个整数 N_i 的乘积，即 $N=N_1\cdot N_2\cdot\cdots\cdot N_\nu=\prod_{i=1}^{\nu}N_i$。

则可以把 N 点 FFT 分解为 ν 段，第 i 段的乘法次数为 $N_i^2 \cdot N / N_i = N_i \cdot N$ 次。把 ν 段的运算次数加起来，再加上每两段之间的 N 次与旋转因子的乘法，总共的乘法次数为

$$\sum_{i=1}^{\nu} N_i N + (\nu - 1) N = \left(\sum_{i=1}^{\nu} N_i + \nu - 1 \right) N \tag{4.2.19}$$

例如，如果 $N=256=16^2=4^4=2^8$，则按（4.2.19）式计算的结果如表 4.2.1 所示。

表 4.2.1　分段数目和乘法次数的关系

分段数目	不分段 $N1=256$	分两段 $N1=16$	分四段 $N1=4$	分八段 $N1=2$
乘法次数	$256N$	$(32+1)N$	$(16+3)N$	$(16+7)N$

从这里可以发现，虽然一般而言，把 FFT 运算的基数取得小，可以得到比较高的运算速度，但也不是愈小愈好。基数到了四以下，速度反而略有降低。所以在系统简单而对运算时间要求苛刻的嵌入式数字处理系统中，常会选用基四 FFT 算法。其实对于基二 FFT，因为 $W_2^0 = 1$，$W_2^1 = -1$，它的算式可以表示为

$$\begin{bmatrix} X(0) \\ X(1) \end{bmatrix} = \begin{bmatrix} W_2^0 & W_2^0 \\ W_2^0 & W_2^1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \end{bmatrix}$$

而因为 $W_4^0 = 1$，$W_4^1 = \mathrm{j}$，$W_4^2 = -1$，$W_4^3 = -\mathrm{j}$，$W_4^{4l} = 1$，四点 FFT 的算式可以表为

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \mathrm{j} & -1 & -\mathrm{j} \\ 1 & -1 & 1 & -1 \\ 1 & -\mathrm{j} & -1 & \mathrm{j} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

注意到这个两种情况下矩阵中的所有乘子大小都是 1，与 x 相乘时，实际上都可以不必用乘法程序，只要按逻辑判断的方法，区别它的正负号和虚实数，即可作相应处理。因此（4.2.19）式中的求和号下的乘法数可以不算，这是基二和基四算法特有的好处。再考虑到二的幂次可以利用蝶形运算节省一半乘法，于是对基二和基四 FFT，其乘法次数分别为（4.2.20）、(4.2.21)式所示。

$$0.5N(\nu - 1) = 0.5N \log_2 N \tag{4.2.20}$$

$$0.5N(\nu - 1) \approx 0.5N \log_4 N \tag{4.2.21}$$

可见，基四 FFT 的乘法次数比基二和 FFT 要少。可以用类似于基二算法那样画出基四算法的流程框图，也可以找到它的排序方法，这留给读者作为习题思考。

4.2.4　其他 FFT 算法

上面介绍的 FFT，它的第一步是把时域数据重新编排，所以称为时域抽取（Decimat-In-Time，DIT）算法。它的对偶算法是让时域数据保持原有的排列，而通过 FFT 运算后，得到一组重新排序了的频域数据。称为频域抽取法（Decimat-In-Frequency，DIF），它的思路和时域抽取是相似的，搞清一种方法就够了。

上两节中讨论的 FFT 只是许多高效率计算 FFT 的方法中的两种，它的基本思想是把一个长为 N 的序列分解成多个最短序列 $N = N_1 N_2 \cdots N_\nu$，其中 N_i（i=1，2，…，ν）为素数，然后开发出相应的高效算法。现在已经有了很多种高效的 FFT 算法，大部分是用 C 语言写的，并且都经过了周密的测试并加上了很多查错功能，使它们能方便地应用。例如 MATLAB 中的 FFT 就可以取任何的 N，不管它是不是 2 的幂。因此编一个好的 FFT 程序要靠专门人才，不

是初出茅庐的生手能干得了的。

FFT 反变换：它和正变换是互成对偶的并有相似的形式如（4.2.22a）式所示。

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-kn}\qquad 0\leqslant n\leqslant N-1 \tag{4.2.22a}$$

取（4.2.22a）式的复共轭，如（4.2.22b）式所示。

$$x^*(n)=\left[\sum_{k=0}^{N-1}\frac{X(k)}{N}W_N^{-kn}\right]^*=\sum_{k=0}^{N-1}\frac{X^*(k)}{N}W_N^{kn} \tag{4.2.22b}$$

可见这是$\frac{X^*(k)}{N}$的傅里叶变换，可以直接调用 FFT 函数来求反傅里叶变换 IFFT。因此，只要做少许修改，用同样的 FFT 算法，就可用来计算 DFT 和 IDFT，今后，如果我们提到 FFT，就同时包括了 FFT 和 IFFT。

在使用 MATLAB 时，就用不到在头脑中绕弯子了。记住用函数 fft 计算 DFT，而用 ifft 计算 IDFT 就行了。注意它们都可以有两个输入变元 x 和 M，若 x 有 N 点，则 fft (x) 或 fft (x, N) 计算它的 N 点 DFT，如果 $N>M$，则函数 fft（X，M）只计算 x 中前 M 点的 DFT。如果 $N<M$，fft（x，M）把 x 延伸 M−N 个零后计算它的 DFT，如果 N 等于 2 的幂，则它的计算效率最高。但如果 N 不等于 2 的幂，计算效率就大打折扣。例如令 x = rand n（1,2^20−1），这个序列包含了 1048575 个样本，它不是 2 的幂。键入 fic，fft（x）；toc 在作者的计算机上所得结果是 4.7s，然而 tic，fft（x, 2^20）；toc 却得到 0.97 s。可见计算 1 048 576 点的 FFT 所需的时间比计算 1 048 575 点 FFT 的时间少了 80%。所以在实际应用时，只要有可能，宁可把 N 选成 2 的幂。

DSP 处理器中的 FFT：FFT 进行的是重复性的乘积累加计算（Multiply Accumulate Calculation，MAC），普通 PC 机用的通用处理器（如 Intel 的 Pentium）需要用许多个指令周期才能完成一个 MAC。所以开发了专门的 DSP 处理器，以便高效地实现 MAC 运算和其他数字信号处理中常用到的运算。例如最新的 DSP 处理器可以在一个指令周期内完成一条 MAC，并且在硬件中实现位倒序，这使得 FFT 的速度有了明显的提高。目前 16 位或更高位的定点 DSP 处理器以及 32 位的浮点 DSP 处理器就已经在市场上可以买到了。

从 FFT 于 20 世纪 70 年代出现，在 20 年左右的时间内，得到了飞速的发展。出现了多种多样的方法，它们各适用特定的情况，可以取得更好些的效果。反映在数字信号处理教材中，FFT 的篇幅也越来越大，一般都单独列成一章。但近 10 年来，国外的教材对 FFT 的介绍篇幅却变得越来越少，一般只有一节。其原因大概是因为 FFT 已非常成熟。在所有的软件系统中，FFT 都是可供调用的标准函数；在所有的 DSP 芯片开发系统中，FFT 也都是标准的固件。一般初学者没有自己编写 FFT 程序的必要，自己编的也必然远比不上系统中的原有程序。所以重要的是会应用系统中的 FFT 程序来解决实际问题。因此，本书减少了讨论 FFT 算法的篇幅，把重点放在 FFT 的应用上。

4.3　用 FFT 计算离散时间序列的频谱

4.3.1　有限长离散时间序列的频谱计算

本节讨论如何用 FFT 来计算有限长序列信号的频谱。考虑一个序号从 n_1 到 n_2 的有限序列 $x(n)$，它的频谱 $X(\omega)$ 定义为它的离散时间傅里叶变换，并且在奈奎斯特频率范围内有界并连续。令序列的长度为 N，$N=n_2-n_1+1$，则计算 $x(n)$ 的 DFT 可以得到 $X(\omega)$ 的 N 个样本点 $X(\omega_k)$。在计算 DFT 时，数字频率 ω_k 的位置由(4.3.1)式计算。

$$\omega_k = k(2\pi/N) = k\mathrm{d}\omega \tag{4.3.1}$$

其中 $k=-(N-1)/2:(N-1)/2$（见 3.2.28 式）。

许多情况下，实际问题要求计算出它的以模拟频率为横坐标的频谱。这时（4.3.1）式应改为

$$\Omega_k = k(2\pi/NT) = kD \tag{4.3.2}$$

其中 D 是模拟域的频率分辨率或频率间隔。$\mathrm{d}\omega$ 则是数字频率的分辨率，$D=F_s\mathrm{d}\omega=\mathrm{d}\omega/T$。在计算 fft 时都是用数字频率，只是在最后画图时，横坐标应该用 kD 来取代 $k\mathrm{d}\omega$。

可以用 FFT 来高效地计算 DFT。在使用 FFT 时，不能用 $n=n_1$ 到 n_2 的 $x(n)$，必须用 $\tilde{x}(n)$ 中主值区间 $n=0, 1, \cdots, N-1$ 的数据，函数 fft($\tilde{x}$,N) 会产生 N 个数据，它们应该定位在频点 $\omega_k = k\mathrm{d}\omega,\quad k=0,1,\cdots,N-1$ 上。如果要求它们处在奈奎斯特频率范围内，可以利用 fftshift(X) 函数。然后这 N 个输出数据就定位于(3.2.28)式中的 k 所决定的频点上，它们就是在 $[-\pi/T,\pi/T)$ 区间内 N 个频点上的频谱 $X(\omega_k)$，因为 $X(\omega)$ 是对所有的 ω 定义的，应该从这 N 个频点样本点进行内插，得出全部的 $X(\omega)$，下面给出几个实例。

例 4.3.1：考虑长度为 5 的有限序列，设采样周期为 0.5 s。

$x(0)=1, x(1)=3, x(2)=5, x(3)=3, x(4)=1$。要求用 FFT 来计算其频谱。

解：因为给出了采样频率，显然要求出的是模拟频率域中的频谱。用 fft 函数求出 x 的 DFT,同时又求出 x 的 DTFT，把它们的幅频和相频特性画在一张图上进行比较。由此编程为程序 hc431：

```
x=[1,3,5,3,1];nx=0:4;T=0.5;                    % 给定原始数据
N=length(x); D=2*pi/(N*T);                     % 求出序列长度及频率分辨率
k=floor((-(N-1)/2):((N-1)/2));                 % 求对称于零频率的 FFT 位置向量
X=fftshift(fft(x,N));                          % 求对称于零频率的 FFT 序列值
subplot(1,2,2),plot(k*D,abs(X),'o:')           %画幅频特性图
subplot(1,2,2),plot(k*D,angle(X),'o:')         % 画相频特性图
```

程序的前面四行用来计算 DFT，并使它的频率范围为对称于零频率点的奈奎斯特区域。五、六两句是用来生成连续的频谱，以便与用 FFT 计算的频谱作比较。程序给出了图 4.3.1 中虚线所示的幅频和相频特性，实线是对所有在奈奎斯特频率范围内的 ω 的连续频谱，而五个空心圆点则是 FFT 计算的结果，当 N 是奇数时，频点对称分布，最左最右边的点在 ±5.03rad / s；如果 N 是偶数，这些圆点是在 $[-6.28,6.28)$ 频率范围内等间距分布的。左边是闭区间，最左的点在−6.28 rad / s 处，右边是开区间，最右的点在 6.28−D 处。FFT 确实计算了频谱中的样本点，而 MATLAB 中的 plot 函数通过在相邻的点之间以直线(图上虚线)连接，用直线进行了插值。图 4.3.1 中各子图的五个圆点和它们之间的连线是 N=5 的结果。

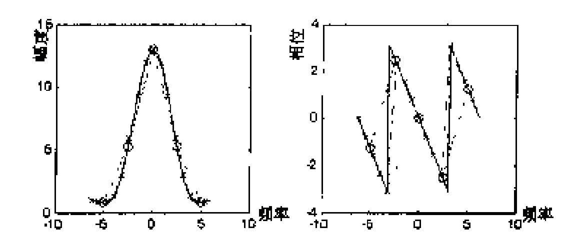


图 4.3.1　用 FFT 计算离散时间序列的频谱，o—不补零；×—补 11 个零；实线—补 1019 个零

4.3.2 用补零方法由 FFT 求 DTFT

对有限序列做 FFT 计算了频谱上有限的样本点，其频率样本的数目必须等于输入数据样本的数目，上例中由于 N=4，所以相邻两个频率样本点的间距为

$$D=\frac{2\pi}{NT}=\frac{2\pi}{5\cdot 0.5}=2.5133\ \ \text{rad/s}$$

这个频率间距非常大，也就是分辨率很差，所以用直线插值所得的幅频和相频特性与实际的特性差别很大。可以用插值方法来提高分辨率，但不方便。由于有了 FFT，提高计算量已经不是问题，改善分辨率的最好方法是增加数据的长度 N。如果没有更多数据，也可以给输入序列补零。例如，在上例中可以在序列后补 11 个零，使序列长度成为 16，则输出的频率分辨率就为 0.39 rad / s，其结果在图 4.3.1 中用“×”号和相连的点划线表示，程序不需做什么改动，只要把 N 值由 length(x)改成 16 就行。因为在使用函数 fft(x, N)时，如果 x 的长度小于 N，它会自动地把 x 后补零达到长度为 N。在图中可以看出频率分辨率有了明显的改善，插值的结果也比较接近于实际情况。如果设 N=1024，得出的是图中的实线，它几乎是连续的频谱了。

上面说到，增加序列长度就能改善频率分辨率，这是基本的规律。但增加数据长度有两种方法，一是真正增加有效数据，它增加了输入序列的阶次，从而提供了频谱的更多细节，这是真正的分辨率。第二种方法是只补零而不增加数据，输入序列和它的频谱阶次依旧没有提高，只是把频谱画的密一些，所以改善的只是图形的视在分辨率，并不能得到频谱的细节的信息。希望读者注意两者的区别。

例 4.3.2：考虑长度为 11 的矩形窗函数 wd(n) 序列，它的频谱函数可以用解析函数表示，见（3.2.25）式和图 3.2.2。要求用 FFT 来计算其频谱。

解：由于序列是实的偶函数。假如选 N=20 作为重复周期，则要在序列后面补个零，在使用 DFT 时，可以把这些零全补在序列的后面，从而计算 n =-5 到 14 的 $w_d(n)$ 的频谱。然而，使用 FFT 时，必须使用按 N=20 的周期延拓所得序列中从 n=0 到 19 的主值部分，因此 FFT 的输入为 $x=[1,1,1,1,1,1,\text{zeros}(1,9),1,1,1,1,1]$。

这也就是周期延拓后的窗函数的前 20 个点，n=[-5:-1]的五个点被移到 n=[15:19]去了。在编写程序时，要准备给出不同的 N 进行比较。下面的程序能适应 N 取 20 和 1024 两种情况，注意连两张子图的位置都在循环中做了安排，读者也要注意这类编程技巧。程序 hc432 的语句如下：

```
C=[20,1024];T=0.5;
for r=[1,2]
N=C(r);D=2*pi/(N*T);
x=[ones(1,6),zeros(1,N-11),ones(1,5)];      % x在主值区间的数组
```

```
k=floor(-N/2+0.5:N/2-0.5);               % 建立对称于零频的频率位置向量
X=fftshift(fft(x,N));                    % 求出 x 的 FFT 并移位至对称位置
subplot(1,2,r),plot(k*D,X)               % 对不同的 N，分别画实频特性不同子图
end
```

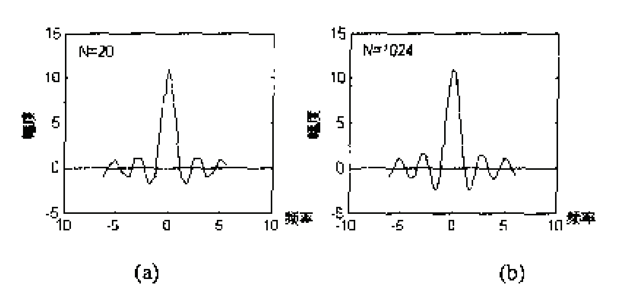


图 4.3.2　用 FFT 补零长度不同所得的矩形窗函数的频谱图

计算的结果见图 4.3.2。注意由于 X 是复数，plot(k*D,X)语句中，只能取 X 的实部，所以实际上和 plot(k*D,real(X)) 等同。图中可见，N=20 虽然给出了频谱的大体形状，但分辨率还是太差，N=1024 的结果画在图 4.3.2 b 上，它和精确的结果已经难以区分了。

在上例中，如果起点不取在 n=0，而取在 n=-5 处，把 x1=[ones(1,11),zeros(1,N-11)]作为 FFT 的输入，计算的将是移位后的窗函数 $\mathrm{wd}(n-5)$，正如第 3 章指出的，时间平移不会影响频谱的幅频特性，它只会在相频特性中引入一个线性相位-5 ω。读者可以自行证实这一点。

归纳前两节的内容，长度为 N 的有限序列 $x[n]$ 的频谱可以用 N 点 FFT 方便地求出。存在的惟一的问题是：频谱是定义在奈氏频率范围内所有的 ω 上的，而 FFT 只计算频率分辨率为 $\mathrm{d}\omega=2\pi/N$ 的样本点上的值。因此，要由插值才能求出全部点的频谱。较简单的插值方法是在输入数据的尾部补零，以改善频率分辨率。如果一个有限序列位置向量不在主值区间，比如对某些 $n<0, x(n)\neq 0$ 的情形，则这些负时间的部分必须移到补零后的尾部，使位置向量位于[0:N-1],然后再用 FFT，否则得到的相频特性将是错误的。

4.3.3　无限长序列的频谱计算

本节将讨论如何用 FFT 计算无限长序列的频谱。上面指出，对于有限长序列，有时必须用补零或者插值来改善频率分辨率，对无限长序列，这个问题不存在了，因为我们可以把数据点取得任意多。然而将遇到另一个问题。在计算机上计算时，总必须把一个无限序列截断成为一个有限序列，而截断会导致泄漏和波动。因此截断的序列只能近似表示无限序列的频谱，这就引发了误差问题。下面用一个实例来说明。

例 4.3.3：　考虑一个采样周期为 0.5 秒的无限序列

$$x(n)=\begin{cases}0.7^n & n\leqslant 0\\ 0 & n<0\end{cases}\tag{4.3.3}$$

现在要问，应该截取多长的序列，才能使计算出的频谱误差小于 1%?

解： 这序列的频谱已经解析地算出为 $x(\omega)=1/(1-0.7e^{-j0.5\omega})$。可以取不同的 N，把计算结果与解析解之差的最大绝对值作为误差 e，把误差和序列的最大幅度之比得到的相对误差作为精度指标。下面的程序 hc433 即取 N=32，求出以百分数表示的相对误差 pe。

```
T=0.5; N=32; D=2*pi/(N*T);
n=0:N-1;x=0.7.^n; k=n;
X=fft(x);                                % 求出 32 点数据的 DFT 数值解
Xt=1.0./(1-0.7.*exp(-j*0.5.*k*D)));      % 求出解析法的 32 点频谱
e=max(abs(abs(X)-abs(Xt)));              % 求数值解和解析解的最大误差
Xm=max(abs(X));                          % 求序列的最大幅度
pe=e/Xm*100                              % 求出最大相对误差（百分数）
```

算出的最大频谱幅度为 Xm = 3.3333，注意 X 的下标有时要加 1，这是因为 MATLAB 中下标必须从 1 而不是从 0 开始。程序算出在 n=0:31 内的 32 个样本点中最大误差为 3.681 410^{-5}。算出的百分数误差 pe 是 0.0011,相当于 0.0011%，或十万分之一。因此对这个序列，取 32 个数据就够了，可见对无限长序列未必需要取很大的 N 来计算频谱。

现在提出这样的问题，要使计算频谱与准确频谱峰值的相对误差不超过 1%， N 值最小应选多少。实际问题和书本例题的不同在于，我们根本不知道频谱的解析解。在准确频谱是未知的情况下，这个问题无法回答；而如果准确频谱已经知道了，又何必去算它呢？所以要用另一种方式提出问题，我们逐次增加 N 值，比较相邻两次计算结果的误差，以这个误差作为判断的标准。要寻找最小的 a 值，使得用 $N=2^a$ 个数据点计算出的频谱，与用 $N/2$ 个数据点的计算结果的误差小于峰值的 β %，比较频谱的误差必须在相同的频点上进行。先讨论比较 $N_1=N$ 和 $N_2=2N$ 的计算频点。对于一个采样周期为 T 的无限序列，用 N_1 点算出的频谱 X_1 位于下列频点上

$$\omega_1(k_1)=k_1\frac{2\pi}{N_1T}\qquad k_1=0,1,\cdots,N_1-1 \tag{4.3.4}$$

而用 N_2 点算出的频谱 X_2，则位于下列频点上

$$\omega_2(k_2)=k_2\frac{2\pi}{N_2T}=k_2\frac{\pi}{N_1T}\qquad k_2=0,1,\cdots,2N_1-1 \tag{4.3.5}$$

令 $\omega_1(k_1)=\omega_2(k_2)$，解出 $k_2=2k_1$。因此必须在 $\omega_1(k_1)=\omega_2(2k_1)$ 的这些频点上比较 X_1 和 X_2 的幅度。因为幅度是偶函数，只需要在 $0\leqslant k_1<N_1/2$ 范围内比较。如果 $N_2\neq 2N_1$，算出的 X_1 和 X_2 将很难找到相同的频点。对它们不好做比较，所以要按 $N_2=2N_1$ 编出程序。

```
% 程序 hc434
T=0.5; a=1; b=100; beta=1;                    % 给定初始数据
while b>beta                                  % 判断是否应结束循环运算
   N1=2^a;n1=0:N1-1;                          % 确定数据长度 N1
   x1=0.7.^n1; X1=fft(x1);                    % 求长度 N1 的序列 x 及其 FFT X1
   N2=2*N1; n2=0:N2-1;                        % 数据长度加倍为 N2=2*N1
   x2=0.7.^n2;X2=fft(x2);                     % 求长度 N2 的序列 x2 及其 FFT X2
   k1p=0:N1/2-1;    k2p=2*k1p;                % 确定两序列对应点的下标 k2=2k1
   d=max(abs(X1(k1p+1)-X2(k2p+1)));           % 求对应点上 FFT 的误差
   mm=max(abs(X1(k1p+1)));                    % 求 X1 幅特性的最大值
   b=d/mm*100;                                % 求相对误差的百分数
   a=a+1;                                     % 序列加长一倍
end
N2,b                                          % 结束循环后显示达到要求的长度 N2 和相对误差 b
k=floor(-N2/2+0.5:N2/2-0.5);D=2*pi/(N2*T);          % 绘图用的奈奎斯特频率范围
子图分割，坐标设定及标注语句省略，只给出两条绘图语句。
plot(k*D,abs(fftshift(X2))),title('(a)振幅频谱')    % 画幅特性
plot(k*D,angle(fftshift(X2))),title('(b)相位频谱')  % 画相特性
```

程序中的 while 循环用来搜索最小的 $N_1=2^a$，使得其计算结果和前一次的误差小于值幅度的 β%，程序中的其余部分就用这个 N_1 和 $N_2=2N_1$ 来计算序列的振幅和相位频谱，并算出其幅特性的最大相对误差 b。如果 $\beta=1$，程序将得到 $N_2=32$，b=0.3323，满足了 $b<\beta$ 的要求。应该说，这时的 b 基本上反映的是 $N=N_1=N_2/2=16$ 时的误差，因为 $N=16$ 时的真正误差比 $N=32$ 时的真正误差要大一个数量级以上。所以这种选择是偏于保守或安全的。其实在选择过程中既然已经按 $N=32$ 算出了较精确的结果，也没必要再退回去取 $N=16$ 重算了。

为了看出选择的过程，现把每次循环所用的序列长度及其对应的相对误差列成表 4.3.1。

从表中可以看出，随着序列的加长，精度呈几何级数地提高。其实只要把程序中的 beta 设得很小，例如设 beta=10^−5，程序就会把 N_2= 4, 8, 16, 32, 64, 128 的情况都计算出来。各次计算出的幅度谱和相位谱见图 4.3.3 中。可以看出，随着序列的加长，曲线逼近的优度也迅速改善。N_2= 16 和 N_2= 32 的曲线已很难分辨，可以想像，它们应该已很接近于无限序列的频谱。

表 4.3.1　取不同的序列长度所对应的相对误差

N_1 : N_2	2 : 4	4 : 8	8 : 16	16 : 32	32 : 64	64 : 128
两者误差 b / %	49	24.0100	5.7648	0.3323	0.0011	1.22e−008

注意图 4.3.3 中 N_2 较小时的曲线不太光滑，这是由于 plot 命令中用了直线插值的缘故，相位谱在 2π 处也有一个缺口，这是因为频谱范围右端是开边界，最右点与边界有一个间距 D，$D=\frac{2\pi}{NT}$。若要减小这个间距，可把 N_2 取 64 或 128。

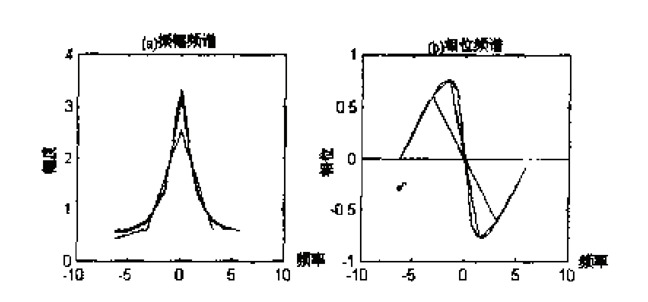


图 4.3.3　N_2 取 4,8,16,32 点时计算出的幅频和相频特性变化

4.4　连续信号采样中的频谱变换

4.4.1　时域采样定理(Nyquist 定理)

为了利用 FFT 等适合于计算机计算的工具来分析模拟信号的频谱，需要弄清经采样后的序列 $x(nT)$ 与原始的模拟信号 $x_a(t)$ 在时域和频谱方面的数学关系。在图 4.1.1 中，我们把对模拟信号进行采样的过程看做使模拟信号通过一个电子开关 S。设电子开关每隔周期 T 合上一次，每次合上的时间为 τ ($\tau \ll T$)，在电子开关输出端得到其采样信号 $x(t)$。用这样的模型，输出序列的幅度与模拟信号相同，而其宽度因而其强度只是模拟信号的 τ/T 倍。所以是衰减了的。要使采样过程中信号不衰减，应该用乘法器代替带阻开关，如图 4.4.1(b)所示。把采样器看作一个乘法器，它的一个输入端接模拟信号，另一个接采样信号。采样信号是宽度为 τ，周期为 T、面积为 1（因而幅度为 1/τ）的矩形脉冲串 $p_\tau(t)$，输出信号 $x(t)$ 就是 $x_a(t)$ 与 $p_\tau(t)$ 相乘的结果。它的宽度虽然减小了，但幅度被放大了。如果让电子开关合上时间 τ→0，则放大倍数趋于无穷，形成理想采样，此时输出信号变成单位冲激串 $p_\delta(t)$

$p_\delta(t)$ 可以表示为：

$$P_\delta(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT) \tag{4.4.1}$$

因此 $x(t)$ 可写成：

$$x(t)=x_a(t)\cdot P_\delta(t)=\sum_{n=-\infty}^{\infty}x_a(t)\delta(t-nT) \tag{4.4.2}$$

其中 $\delta(t)$ 是单位冲激信号，在上式中只有当 $t=nT$ 处才可能有非零值，因此写成式(4.4.3)。

$$x(t)=x_a(t)\cdot P_\delta(t)=\sum_{n=-\infty}^{\infty}x_a(nT)\delta(t-nT) \tag{4.4.3}$$

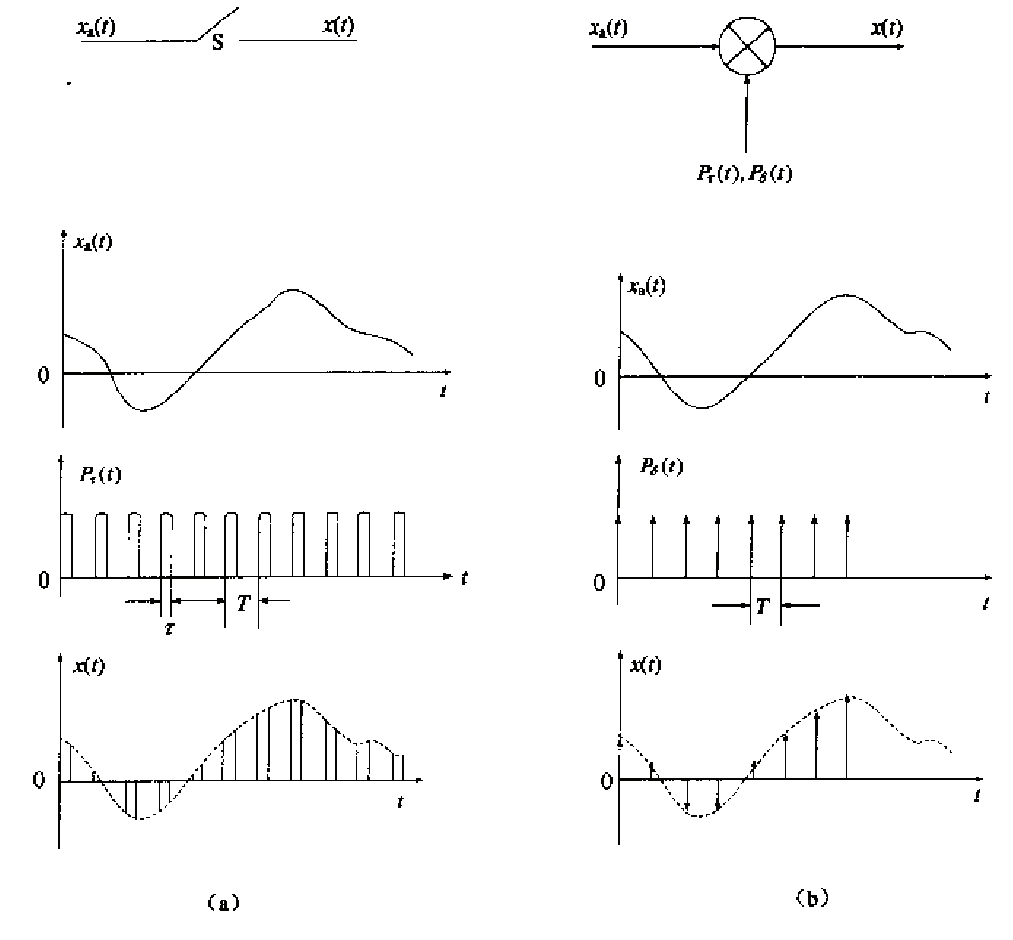


图 4.4.1　对模拟信号进行采样

这就是理想采样前后时域信号的数学关系，对两边进行傅里叶变换，就可以找到它们之间在频域的关系。根据傅里叶变换的性质，两信号在时域乘积的傅里叶变换等于它们各自的傅里叶变换的卷积，如果 $X_a(\mathrm{j}\Omega)=F[x_a(t)]$，$X(\mathrm{j}\Omega)=F[x(t)]$ 根据冲激函数 $\delta(n)$的性质（见表 3.2.1）：

$$F[\delta(t)]=\int_{-\infty}^{\infty}\delta(t)\mathrm{e}^{-\mathrm{j}\Omega t}\mathrm{d}t=1 \tag{4.4.4}$$

$$P_\delta(\mathrm{j}\Omega)=F[\mathrm{p}_\delta(t)]=\sum_{k=-\infty}^{\infty}2\pi\, a_k\delta(\Omega-\Omega_\mathrm{s}) \tag{4.4.5}$$

$$a_k=\frac{1}{T}\int_{-T/2}^{T/2}\delta(t)\mathrm{e}^{-\mathrm{j}k\Omega_s}\mathrm{d}t=\frac{1}{T} \tag{4.4.6}$$

式中 $\Omega_\mathrm{s}=2\pi/T=2\pi F_\mathrm{s}$ 为采样角频率，单位为 rad / s。因此

$$X(\mathrm{j}\Omega)=\frac{1}{2\pi}X_\mathrm{a}(\mathrm{j}\Omega)\otimes P_\delta(\mathrm{j}\Omega)=\frac{1}{2\pi}\cdot\frac{2\pi}{T}\int_{-\infty}^{\infty}X_a(\mathrm{j}\Omega)\sum_{k=-\infty}^{\infty}\delta(\Omega-\Omega_s-\theta)\mathrm{d}\theta$$

$$=\frac{1}{T}\sum_{k=-\infty}^{\infty}\int_{-\infty}^{\infty}X_a(\mathrm{j}\Omega)\delta(\Omega-\Omega_s-\theta)\mathrm{d}\theta$$

最后得到

$$X(\mathrm{j}\Omega)=\frac{1}{T}\sum_{k=-\infty}^{\infty}X_\mathrm{a}(\mathrm{j}\Omega-\mathrm{j}k\Omega_\mathrm{s}) \tag{4.4.7}$$

上式表明，采样信号的频谱是原模拟信号的频谱无限次平移的叠加。每次平移的间隔为采样角频率Ω_s。因为模拟频率是沿正负方向无限延伸的，模拟信号的频谱也可以沿正负方向无限延伸，而采样信号的频谱却限制在$[-\Omega_s/2,\Omega_s/2)$的奈奎斯特频率区间（见图 3.1.4)。这是一个多点对一点的映射，因此从这个意义上也可以理解，把模拟频谱映射到数字频谱的区间来，就必然要进行周期性的平移叠加。

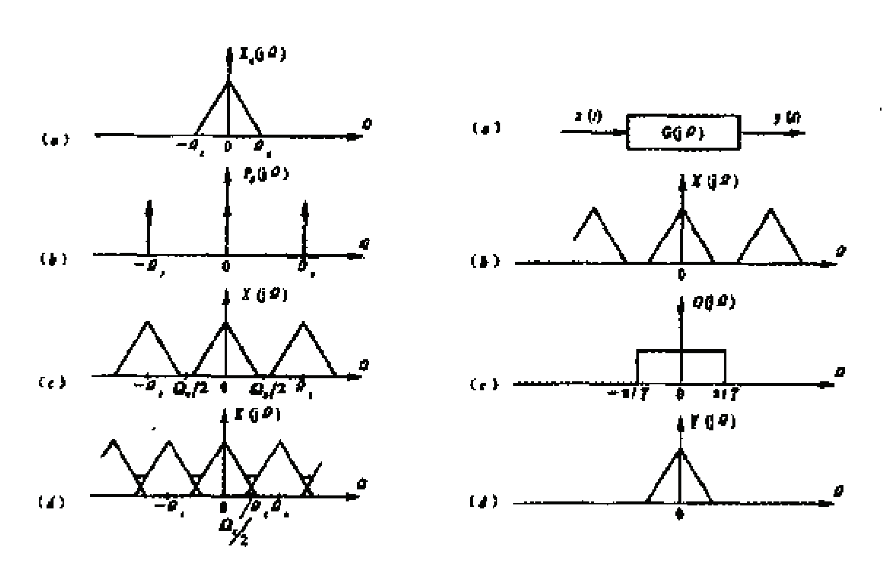


图 4.4.2　采样信号的频谱　　　　图 4.4.3　采样恢复

设模拟信号是带限信号，最高频率为Ω_c，其频谱$X_a(j\Omega)$如图 4.4.2 (a)所示。$p_\delta(t)$的频谱$P_\delta(j\Omega)$用图 4.4.2 (b)表示。那么按照（4.4.7）式，$x(t)$的频谱用图 4.4.2 (c)表示。图中相当于原模拟信号的频谱称为基带频谱。如果基带频谱较窄，满足$\Omega_c<\Omega_s/2$，或者$f_c<F_s/2$，基带谱与它的周期延拓形成的谱没有重叠，此时的数字信号频谱就如图 4.4.2 (c) 所示的周期频谱。如果让这个采样信号经过理想低通滤波器得到输出$y(t)$，就能从中不失真地提取原模拟信号，如图 4.4.3 所示。在此条件下，(4.4.7)式将成为：

$$Y(j\Omega)=\frac{1}{T}X_a(j\Omega) \tag{4.4.8}$$

即采样序列的频谱等于原始模拟信号的频谱除以采样周期 T。

但如果选择采样频率Ω_s低，或者说信号最高频率Ω_c高，以致$\Omega_c>\Omega_s/2$，或者$f_c>F_s/2$，则基带谱与把它做Ω_s的周期延拓形成的频谱就会发生重叠，称为频谱混叠现象，如图 4.4.2(d)所示。这种情况下，再用如图 4.4.3 所示的理想低通滤波器对它进行滤波，得到的将是已失真的模拟信号。

需要说明：频谱函数应该复数相加，图 4.4.2 仅是示意图。一般称$\Omega_s/2$为折叠频率，只有当信号最高频率不超过该频率时，才不会产生频率混叠现象，否则超过$\Omega_s/2$ 的频谱会折叠回来形成混叠现象，因此最大的频率混叠产生在$\Omega_s/2$（或$F_s/2$)附近。

总结上述内容，可把采样定理叙述如下：

（1）对连续信号进行等间隔采样形成采样信号，采样信号频谱是原连续信号的频谱以采样频率为周期进行周期性的延拓并叠加形成的，用公式（4.4.7）表示。

（2）设连续信号$x_a(t)$属带限信号，其带宽为Ω_c，如果带宽小于折叠频率，$\Omega_c<\Omega_s/2$，那么让采样信号$x(t)$通过一个增益为T，截止频率为$\Omega_s/2$的理想低通滤波器，可以完全恢复原

始的模拟信号 $x_a(t)$。如果不满足上述条件，即 $\Omega_s/2<\Omega_c$，就会造成采样信号中的频谱混叠现象，从这样的采样信号中不可能无失真地恢复原连续信号。上述条件也称为奈奎斯特条件。

实际中对模拟信号进行采样，需根据模拟信号的带宽，按照采样定理的要求选择采样频率 F_s，但考虑到任何有限长度模拟信号的频谱都不是锐截止的，带宽频率以上还有较小的高频分量。为了减小这些高频分量对基带信号的混叠，通常在采样之前加一保护性的低通滤波器，滤去高于 $F_s/2$ 的有害高频分量以及其他的一些干扰信号。这就是在图 1.5.1 中采样之前加预滤波器的原因。

4.4.2　由离散序列重构模拟信号

由离散序列重构模拟信号是采样的逆过程——从离散到连续。设数字系统给出一个序列信号 $x(n)$，现在要把它恢复为原来的模拟信号 $x_a(t)$。设 $x(n)$的频谱为 $X(\mathrm{j}\Omega)=X(\mathrm{j}\omega/T)$，根据公式（4.4.7），它所携带的模拟频谱具有无限宽的频带。

$$\sum_{k=-\infty}^{\infty}X_a(\mathrm{j}\Omega-\mathrm{j}k\Omega_s)=T\cdot X(\mathrm{j}\Omega) \tag{4.4.9}$$

把这个信号通过一个带宽为$[-\pi/T,\ \pi/T]$的理想滤波器。只把 $k=0$ 的奈奎斯特主值区间的频谱过滤出来，如果信号带宽小于 π/T，就可以完全无失真地恢复原来的低频模拟频谱。频谱给定为

$$X_a(\mathrm{j}\Omega)=T\cdot X(\mathrm{j}\Omega) \tag{4.4.10}$$

因此，由离散序列重构模拟信号的过程可以分解成两步。首先经过一个脉冲变换器，将离散序列乘以 T，使它变成一个等价的模拟信号 $x_s(t)$；然后再让它通过一个低通滤波器。下面分理想方法和实际方法两种情况，分别进行详细讨论。

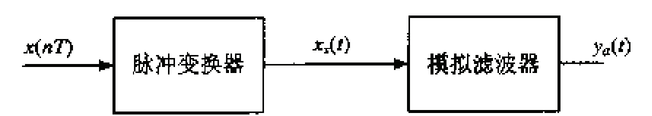


图 4.4.4　由离散序列重构模拟信号的等效框图

（1）理想方法

首先用脉冲变换器把输入序列 $x(nT)$变换为连续信号，如式(4.4.11)。

$$x_s(t)=T\cdot\sum_{n=-\infty}^{\infty}x(nT)\,\delta(t-nT) \tag{4.4.11}$$

$x_s(t)$是加了权 $Tx(nT)$的脉冲序列时间函数。为什么不能让 $x(nT)$直接去驱动滤波器而一定要先把它变为 $x_s(t)$呢？因为直接把序列 $x(nT)$加到滤波器，是不会有输出的。要驱动一个连续系统，输入信号必须具备能量，或者说，在信号曲线下，应该围有面积。一个只具有数值但宽度为零的序列是不可能驱动连续系统的而把它变成脉冲时间函数 $\delta(t)$后，情况就不同了。因为 $\delta(t)$虽然宽度为零，但它包的面积为 1，这样把离散序列变换为以时间为自变量的信号。为什么还要乘以 T ？这在第 2.1 节就已指出，$x(n)$与它对应的模拟信号 $x_a(t)$量纲不一致，相差一个乘子 T，那是从时域分析的结论。现在从频域分析中的(4.4.10)式，又得知两者的频谱也不等价，也差一个时间量纲——采样周期。可见，把 $x(nT)$变为 $x_s(t)$一是为了离散自变量换成连续自变量；二是把采样序列变成等价序列。这是推动滤波器之前必须先做的步骤。

然后让 $x_s(t)$ 通过一个幅频特性如图 4.4.3 (c)的理想重构滤波器。它在$[-\pi/T, \pi/T]$的奈奎斯特频率范围内，幅特性为 1，即允许所有频率成分无失真地通过。而在$[-\pi/T, \pi/T]$ 区间以外，则阻止任何频率成分通过。即可表示为：

$$H(\mathrm{j}\Omega)=\begin{cases}1 & |\Omega|\leqslant\Omega_c\\ 0 & |\Omega|>\Omega_c\end{cases} \tag{4.4.12}$$

它的脉冲响应可用傅里叶反变换求出，为（参见例 3.2.8）

$$h(t)=\frac{1}{2\pi}\int_{-\Omega_c}^{\Omega_c}\mathrm{e}^{\mathrm{j}\Omega_c t}=\frac{\sin(\Omega_c t)}{\pi t} \qquad -\infty<t<\infty \tag{4.4.13}$$

对于 $\Omega_c=\pi/T$ 的奈奎斯特滤波器，它的脉冲响应（4.4.13 a）就成为

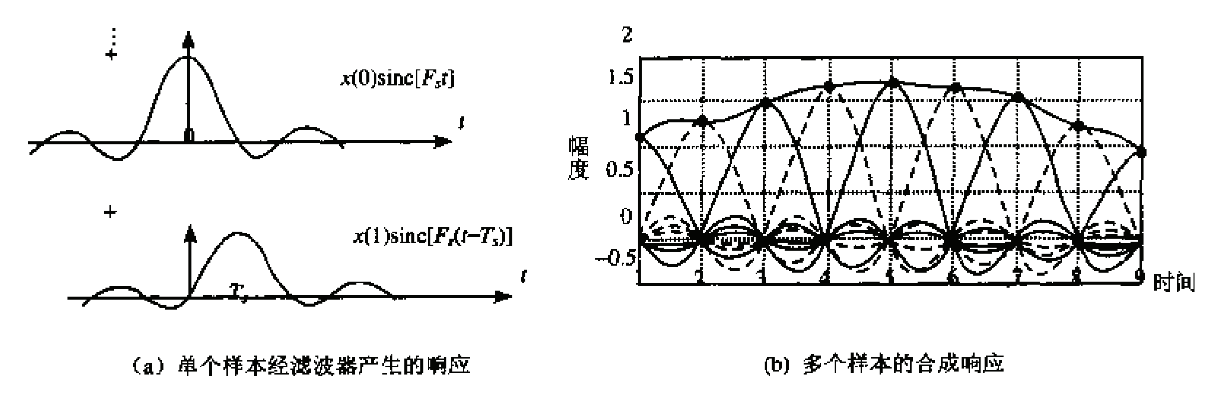
$$h(t)=\frac{\sin(\pi t/T)}{\pi t} \qquad -\infty<t<\infty \tag{4.4.13 a}$$

它在 t<0 的区域不等于零，因此它是非因果的，也是不能实现的。因此，离散序列 $x(n)$ 经过图 4.4.4 的理想重构系统后的输出 $y_a(t)$ 就是 $x_s(t)$ 和（4.4.13a）式中 $h(t)$ 的卷积：

$$y_a(t)=T\cdot\sum_{n=-\infty}^{\infty}x(n)h(t-nT)=\sum_{n=-\infty}^{\infty}x(n)\frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T} \tag{4.4.14}$$

公式（4.4.14）的几何意义可用图 4.4.5 说明。理想滤波器的脉冲响应见图 4.4.5(a)，其中下图是右移一位的响应。当成串的脉冲经过滤波器后，其结果是按（4.4.14）式将它们叠加，像图 4.4.5(b)那样。因为脉冲响应 $h(t-nT)$在 $t=nT$ 点都等于零，叠加的结果对除中心样本点外的其他样本点的值没有影响。所以输出的模拟信号 $y_a(t)$ 将经过所有的样本点 $x(n)$，成为样本序列的包络。也就是说，$y_a(t)$ 是对离散样本点 $x(n)$插值的结果，（4.4.14）式是其插值公式。

必须注意，理想方法的两个步骤都是实际上无法实现的。第一步中的脉冲函数 $\delta(t)$在工程中就无法做到；第二步中的理想滤波器的脉冲响应是一个非因果过程，在输入样本之前要求输出响应已经产生，见图 4.4.5(a)。这也是物理上无法实现的。所以理想方法只有理论意义，它给出了一种理论计算的方法和努力追求的目标。



(a) 单个样本经滤波器产生的响应　　(b) 多个样本的合成响应

图 4.4.5　理想的重构过程相当于插值

（2）实际方法

在实际的系统中，一般采用 D/A 变换器来重构模拟信号。D/A 变换器的芯片结构中，通常都包括了零阶保持器（zero-order-holder），它可以起到脉冲变换器的作用。

- 首先它把数字信号变换为模拟信号，然后，将信号保持一个采样周期 T，直到下一个新的数字信号的到来。这样，就相当于把信号乘了 T，实现了对增益的要求，在物理上构成了一个面积 $Tx(n)$，把离散的序列变成了连续的模拟信号。因此，脉冲变换器的两项任务，它都完成了。它比理想脉冲变换器要差的方面主要是产生了时间的滞后，因为这时序列样本的中心，后移了 T/2，对于正弦信号，这相当于一定的负相移。
- 零阶采样保持器还有一定的低通滤波作用，可以分析如下。它的脉冲响应为

$$h(t)=\begin{cases}1 & 0\leqslant t\leqslant T\\ 0 & t<0 \text{ 及 } t>T\end{cases} \tag{4.4.15}$$

它的频率特性可以用傅里叶变换求得如下：

$$H(\mathrm{j}\Omega)=F[h(t)]=\int_{t=0}^{t}e^{-\mathrm{j}\Omega t}\mathrm{d}t=\frac{\mathrm{e}^{-\mathrm{j}\Omega T}-1}{-\mathrm{j}\Omega}=T\cdot\frac{1-\mathrm{e}^{-\mathrm{j}\omega}}{\mathrm{j}\omega} \tag{4.4.16}$$

T 是它的等效增益，已在上面讨论过。它的频率特性，可以用 hc441 来绘制。

```
w=0:0.1:30;H=(1-exp(-j*w))./(j*w);
subplot(1,2,1),plot(w,abs(H))
subplot(1,2,2),plot(w,angle(H))
```

得出的幅特性和相特性见图 4.4.6 中的实线。图中还用虚线画出了理想滤波器的频率响应（其相特性为零，图中看不清），供读者比较。如果滤波作用还不够，就在 A/D 变换器以后，另加平滑模拟滤波器。这个平滑滤波器的设计，可参阅第 8 章。

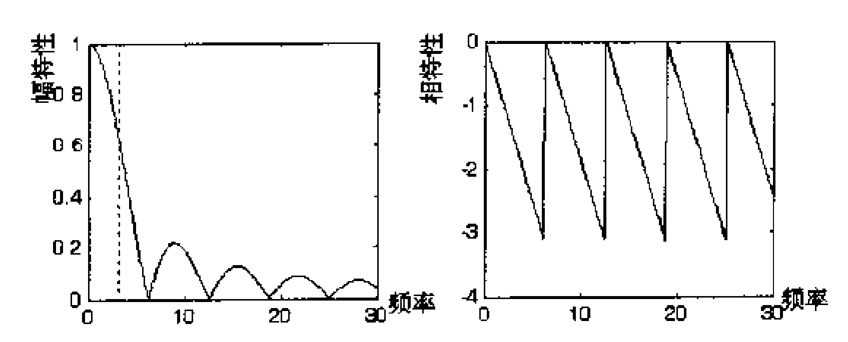


图 4.4.6　零阶保持器（实线）和理想尼奎斯特滤波器（虚线）的幅频特性和相频特性

4.4.3　理想预滤波作用的定量分析

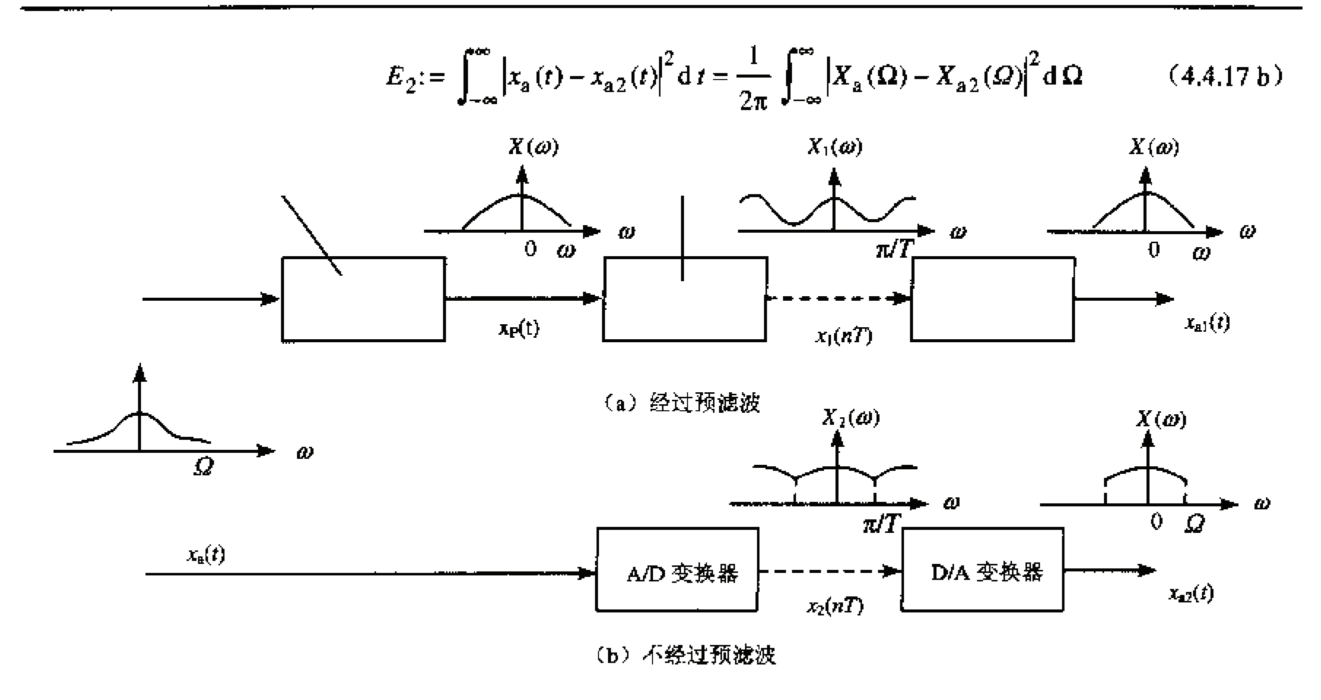
本节将对预滤波的作用进行定量分析。在图 4.4.7 (a)中，连续信号 $x_a(t)$ 经过预滤波后为 $x_p(t)$，然后把它采样后变成 $x_1(nT)$。假定立即再用一个带理想低通滤波器的 D/A 变换器把它转换回连续信号 $x_{a1}(t)$，这是第一种情况。另外在图 4.4.6 (b)中，连续信号 $x_a(t)$ 不经过预滤波而被采样，得到的 $x_2(nT)$ 也立即再用同样的 D/A 变换器把它转换回连续信号 $x_{a2}(t)$。下面比较经过预滤波的 $x_{a1}(t)$ 和不经过预滤波的 $x_{a2}(t)$ 与原始信号 $x_a(t)$ 的差别。

假如原始信号 $x_a(t)$的带宽为 W，且 $W<\pi/T$。而预滤波器的通带等于或大于 W，则 A/D 变换将不引入频率混叠，因而 $x_1(nT)=x_p(nT)=x(nT)=x_2(nT)$，并有 $X(\Omega)=X_1(\Omega)=X_2(\Omega)$。因而有无预滤波器没有差别。

如果 $x_a(t)$ 不是有限带宽，对它进行数字处理会带来误差。此时 $x_{a1}(t)$ 和 $x_{a2}(t)$ 都不会与 $x_a(t)$ 相同。下面引入能量的积分来比较这两个过程。利用帕塞伐尔定理，定义

$$E_1:=\int_{-\infty}^{\infty}\left|x_a(t)-x_{a1}(t)\right|^2\mathrm{d}t=\frac{1}{2\pi}\int_{-\infty}^{\infty}\left|X_a(\Omega)-X_{a1}(\Omega)\right|^2\mathrm{d}\Omega \tag{4.4.17 a}$$

$$E_2 := \int_{-\infty}^{\infty} \left| x_a(t) - x_{a2}(t) \right|^2 \mathrm{d}t = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left| X_a(\Omega) - X_{a2}(\Omega) \right|^2 \mathrm{d}\Omega \qquad (4.4.17\,b)$$



（a）经过预滤波

（b）不经过预滤波

图 4.4.7　经过预滤波(a)和不经过预滤波(b)的连续信号经采样重构后输出的不同结果

为平方积分误差。如果低通预滤波是理想的，则

$$X_p(\Omega) = \begin{cases} X_a(\Omega) & |\Omega| \leqslant \pi/T \\ 0 & |\Omega| > \pi/T \end{cases}$$

因为 $x_p(t)$是有限带宽的，对它采样不会引起频率混叠。因此

$$X_1(\Omega) = X_p(\Omega)/T \quad |\Omega| \leqslant \pi/T。$$

由于 $X_1(\Omega)$ 可以按 $2\pi/T$ 进行周期延拓到全部 Ω，因此理想 D/A 变换器输出 $x_{a1}(t)$的频谱为

$$X_{a1}(\Omega) = \begin{cases} T \cdot X_1(\Omega) = X_p(\Omega) = X_a(\Omega) & |\Omega| \leqslant \pi/T \\ 0 & |\Omega| > \pi/T \end{cases} \qquad (4.4.18)$$

利用这个关系，（4.4.17 a）式中输入 $x_a(t)$ 和输出 $x_{a1}(t)$ 的平方积分误差可以表示为

$$E_1 := \frac{1}{2\pi} \left(\int_{-\infty}^{-\pi/T} \left| X_a(\Omega) \right|^2 \mathrm{d}\Omega + \int_{\pi/T}^{\infty} \left| X_a(\Omega) \right|^2 \mathrm{d}\Omega \right) \qquad (4.4.19)$$

注意其积分的范围恰好是整个奈奎斯特频率范围之外的频段。可见，如果 $x_a(t)$ 是有限带宽，且带宽小于奈奎斯特频率范围时，误差为零。

下面来计算图 4.4.5 (b)中的没有预滤波的情况。由于存在着频率混叠，在$|\Omega| \leqslant \pi/T$ 的范围内，$X_{a2}(\Omega)$ 将不等于 $X_a(\Omega)$，因此

$$X_{a2}(\Omega) \begin{cases} \neq X_a(\Omega) & |\Omega| \leqslant \pi/T \\ = 0 & |\Omega| > \pi/T \end{cases} \qquad (4.4.20)$$

因而

$$\begin{aligned} E_2 &:= \frac{1}{2\pi} \int_{-\infty}^{\infty} \left| X_a(\Omega) - X_{a2}(\Omega) \right|^2 \mathrm{d}\Omega \\ &= \frac{1}{2\pi} \left(\int_{-\infty}^{-\pi/T} \left| X_a(\Omega) \right|^2 \mathrm{d}\omega + \int_{-\pi/T}^{\pi/T} \left| X_a(\Omega) - X_{a2}(\Omega) \right|^2 \mathrm{d}\omega + \int_{\pi/T}^{\infty} \left| X_a(\Omega) \right|^2 \mathrm{d}\Omega \right) \end{aligned} \qquad (4.4.21)$$

把（4.4.19）与（4.4.17a）进行比较，因为它多了奈奎斯特频率范围之内一段的能量积分，

$$\int_{-\pi/T}^{\pi/T}\left|X_a(\Omega)-X_{a2}(\Omega)\right|^2 d\omega>0$$

可以得知 $E_2>E_1$，证实了信号经过预滤波后再做数字处理比不经过预滤波的误差要小。因为这个误差主要取决于采样后造成的频率混叠，所以预滤波器的带宽应当取得比奈奎斯特频率 π/T 小。

4.5　连续信号的频谱计算

连续时间信号的频谱计算，过去知道的方法只有在“信号于系统”中介绍的直接积分法。它的好处是积分可以得出闭合的解析形式，其频谱可以精确地画出。但对于实际中遇到的大多数复杂的信号以及由测量得到的信号，都不能采用这种方法。这时，必须采用计算机辅助的方法。而用计算机计算的第一步就是把连续信号离散化，一是采样；二是截断。所以，不仅是实时处理信号要用数字信号处理的理论，非实时地分析研究大部分连续信号同样要用数字信号处理的理论和方法。所以说，“数字信号处理”不应理解为“对数字信号进行处理的理论和方法”，而应理解为“对信号进行处理的数字方法和理论”。本节即着重讨论用 FFT 计算连续时间信号的频谱问题。

4.5.1　非周期连续信号的频谱计算

设 $x_a(t)$ 为绝对可积的连续信号，并设 $x(nT)$ 是它的采样序列，如果 T 足够小，有

$$X_a(\Omega)=F[x_a(t)]\approx\begin{cases}T\cdot F[x(nT)]=T\cdot X(\Omega T) & |\Omega|\leqslant\pi/T\\ 0 & |\Omega|>\pi/T\end{cases}\tag{4.5.1}$$

数字频谱 $X(\omega)=X(\Omega T)$ 可以用 FFT 求得，所以可由（4.5.1）求得连续信号的频谱。考虑正时间信号，即在 $t\leqslant0$ 的区间内取值全部为零的信号。时刻 $t=0$ 是相对的，可以把开始研究某信号的时刻看做 $t=0$，因此在实际应用中，所有的信号都可看做正时间信号。

先讨论信号时域的长度 L，采样周期 T 和样本数 N 之间的关系，根据 $x_a(t)$ 在边界 $t=0$ 和 L 处有或没有样本，在区间 $(0, L)$, $[0, L]$, $(0,L]$ 和 $[0, L)$（圆括号表示开边界，方括号表示闭边界）中包含的频点数目是不一样的。$x_a(t)$ 在上面四个区间内的样本数将是 $N-1, N+1, N$ 和 N。因此，从它们的样本组计算出的四个频谱也会不同。如果 N 非常大，这四个频谱应该很接近。为了方便起见，以后规定，把时间记录的长度选为 $[0,L)$，在这样的假设下，$L=TN$。

由(4.5.1)式计算 $x_a(t)$ 的频谱，包括如下三个问题：

（1）频率混叠。如果 $x_a(t)$ 不是有限带宽的，则采样周期 T 选择的主要依据是，使由时间采样所造成的频率混叠小到可以忽略。

（2）频率分辨率。$x_a(t)$ 的频谱 $X_a(\Omega)$ 是定义在所有 Ω 上的，当使用 FFT 来计算 $T\cdot X(\omega)$ 时，只计算出了它的 N 个频率样本值 $X(k\cdot D)=X(2k\pi/NT)$，$k=0, 1, \ldots, N-1$。还必须从这些样本值中插值出 $X_a(\Omega)$，当地频率分辨率 D 与 L 成反比，且 D 愈小，插值的结果就愈好。因此即使 $x_a(t)$ 是有限时间的，我们也愿意选择一个大的 $L=NT$ 来改善频率分辨率。

（3）截断效应。如果 $x_a(t)$ 是无限长的，就必须把它截断到长度 $L=NT$，截断会引起吉布斯效应（波动），也会把窗函数的频谱引入信号频谱，造成混叠，所以需要考虑其误差的问题。

由于上述三个问题，当用 FFT 来计算 $x_a(t)$ 的频谱时必须小心。很明显，采样周期 T 应

该选得小些以减少混叠，而 N 要选得足够大来提高分辨率；如果 N 是确定的，为了减小频率混叠就要减小 T；为了减小截断效应，需要增加 $L=NT$，所以要增大 T；这是两个互相矛盾的要求。此外，取很小的 T 和很大的 N 往往又是不必要的。因此建议在计算 $x_a(t)$ 与频谱 $x_a(\Omega)$ 时，采用以下的步骤来选择 T 和 N。

（1）先初步选择时间记录长度 L，使得 0 到 L 之间包括了大部分非零的 $x_a(t)$，然后用逐次减小 T 和加大 N 的步骤来选择周期 T，使得时间采样造成的频率混叠可以忽略不计。方法就是：选择一个较小的 N_1 值。算出 $T_1=L/N_1$，然后用 N_1 点的 FFT 来计算频谱 $X(2\pi k/(N_1T_1))$。将 X 乘以 T_1 并用 plot 命令进行插值，这就得出了 $x_a(t)$ 频谱 $X_a(\Omega_k)$ 的第一次近似值。然后用 $T_2=T_1/2$ 及 $N_2=2N_1$ 重复这一计算。由于两个计算结果是在同样的 L 下得到的，它们具有相同的截断效应，这样两个频谱间的误差都是由于频率混叠引起的。如果两个频谱非常接近，比如说其最大幅度误差小于峰值幅度约 1%，那么频率混叠大体可以忽略。以后的计算中都可采用 T_2 作为采样周期；如果精度还不够，可以另选较小的 T_3, T_4 和较大 N_3，N_4 等。

（2）在选定 T 以后，进一步选择 L。用 T 和 $2N$ 来计算 $x_a(t)$ 的频谱，即把 L 增加一倍。并与用 T，N 的结果进行比较。因为两个频谱所用的 T 相同，它们的奈奎斯特频率范围和频率混叠应该相同。因此，两个频谱之间的任何差别都是由于截断效应不同（L 不同）引起的。如果两者的误差不大，这个 N 就可以接受；如果两者的误差太大，就把 N 加倍，直到用两个相继的 N 算出的频谱非常接近为止。这时得到的频谱应该很接近于 $x_a(t)$ 的真实幅频特性。

如果 $x_a(t)$ 是绝对可积的，那么，用上面步骤算出的幅频特性，应该收敛到 $x_a(t)$ 的真实的频谱 $X_a(\Omega_k)$ 上。这可用下例来说明。

例 4.5.1：考虑连续时间信号 $x_a(t)=e^{-0.1t}$，$(t\geq 0)$，用 FFT 计算其频谱。

解：此信号的峰值幅度为 1 而在 $t\geq 50$ 时其幅度小于 0.0067，因此如果选择 $L=50$，则记录时间长度将覆盖信号的主要部分。先选较小的值 $L=10$ 来更好地说明计算的过程。注意 $e^{-0.1*10}=0.37$，所以 $L=10$ 并没有完全包括信号的主要部分。但时间区间[0,10]还是能大体反映 $t>0$ 的信号的主要频率分量。

先随意地选择 $N=5$，得到 $T=L/N=2$。下面是程序 hc451 的核心语句。

```
T=? ;N=? ;D=2*pi/(N*T);
n=0:N-1;x=exp(-0.1*n*T);          % 给出序列信号，x 与 Xa 在 MATLAB 中相同
Xa=T*fftshift(fft(x));             % 求其 FFT 乘以 T 转换为模拟频谱，移至零频中心
k=floor(- (N-1)/2:((N-1)/2);       % 位置向量也移至零频中心
Xa(1),plot(k*D,abs(Xa))            % 求频谱最左第一个元素(即奈奎斯特频率处)的值，绘图
```

得到的幅频特性画在图 4.5.1 (a)中。然后用 $T=1$，及 $N=10$ 重复计算，所得结果在图 4.5.1(b)上表示。注意两幅图上的频率坐标是相同的，但它们的奈奎斯特频率边界不同。

可以用奈奎斯特频率处的幅特性来评价混叠的严重程度。图 4.5.1 (a)中 $T=2$，奈奎斯特频率边界是 $\left[-\frac{\pi}{2},\frac{\pi}{2}\right)=[-1.57,1.57)$，图中做出了标志；而图 4.5.1 (b)的 $T=1$，奈奎斯特频率边界是 $\pm\pi/1=\pm 3.14$，它们已经超出了横轴的边界。可以看出，即使在图 4.5.1 (b)的边界(± 3)处，仍有一个相当大的非零值，用程序检验-3.14 处的值为 $X_a(1)=0.3319$。说明在 $T=1$ 时频率混叠不能忽略。

继续多次把 T 减半和把 N 加倍（保持 L 不变）进行计算，直到 $T=0.1, N=100$，其结果在

图 4.5.1 (d)画出。这时的奈奎斯特频率已经是$\pm\pi/0.1=\pm31.4$ 1/s，在边界上的 X_a (1)值已减小到 0.0318，因为峰值是 6，相对误差小于 0.032/6=0.4%。这说明，如果 T=0.1，由时域采样引起的频率混叠可以忽略不计。仔细的数值分析也说明，图(c)和图(d)中两根曲线的差别不大，故以后的计算中都用 T=0.1。为了更好地比较，在 4.5.1 中都用同样的模拟频率范围 [-3, 3] 绘图，这可分别用 axis([-3, 3, 0, 10])来实现。

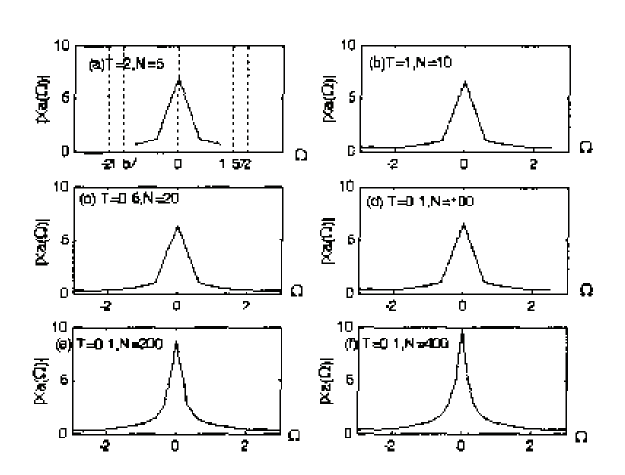


图 4.5.1　取不同的采样周期 T 和数据长度 L=T*N 对频谱计算的影响

选定 T 以后，下面来分析改变 L（也就是改变 N）的效果。图 4.5.1 (e)用实线画出了 T=0.1, N=200 的计算频谱，与图 4.5.1 (d)中 T=0.1, N=100 的结果进行比较。这里只画出了[-3, 3]的频率范围，可见它们之间的区别很大，说明截断效应相当明显，有必要采用大些的 L。图 4.5.1 (f)给出了用 T=0.1, N=400 的计算频谱。与图 4.5.1 (e)相比仍有相当差别。如果再画出 T=0.01, N=800 的结果，它们的误差在[-3，3]频段上就很难分辨了。因此，可以得出结论：在计算 $x(t)=\mathrm{e}^{-0.1t}$ 频谱时，T=0.1 已经小得足以避免频率混叠，而 L=0.1×800=80s 已经大得足以避免截断效应的影响。

为了证实这个论断，可以用（4.5.2）式解析地求出 $x_a(t)=\mathrm{e}^{-\alpha t}$，($t\geqslant0$)的精确频谱。

$$X_a(\Omega)=\int_0^{\infty}\mathrm{e}^{-\alpha t}\mathrm{e}^{-\mathrm{j}\Omega t}\,\mathrm{d}t=\frac{-1}{\alpha+\Omega\,j}\;\mathrm{e}^{-(\alpha+\mathrm{j}\Omega)t}\Big|_{t=0}^{\infty}=\frac{1}{\mathrm{j}\Omega+\alpha}\tag{4.5.2}$$

将 α =0.1 代(4.5.2)式入，得

$$X_a(\Omega)=\frac{1}{\mathrm{j}\Omega+0.1}=\frac{1}{\sqrt{\Omega^2+0.01}}\mathrm{e}^{-\mathrm{j}\arctan(10\Omega)}\tag{4.5.3}$$

可以按(4.5.3)式画出精确的频谱，它与图 4.5.1 (f)中 FFT 计算的最精确的结果来比，差别是很小的。

如何用一个统一的程序而用不同的参数代入，画出图 4.5.1 这样的多幅子图呢，这里给读者介绍一点 MATLAB 的编程技巧。在上述的程序核心语句外围，加一个 for 循环，又把各次计算中要改变的参数，写成数组，然后靠循环指数 r 来依次提取这些参数，也用 r 来指定不同的子图。于是所得程序如下

```
T0=[2,1,0.5,0.1,0.1,0.1];     % 各次计算拟采用的T,编成向量T0
L0=[10,10,10,10,20,40];       % 各次计算拟采用的N,编成向量N0
for r=1:6                     % 循环计算六次
T=T0(r);N=L0(r)/T0(r);        % 根据计算顺序选用T及N
（插入程序hc451 核心语句）
```

```
    subplot(3,2,r),plot(k*D,abs(Xa))  % 在位置r处绘制子图
    end
```

用这种方法时，要注意在核心语句中，必须少用可调参数，使得改动一两个参数就能完成新的计算。比如我们只准备改动 T 和 L,则 N 就不要单独赋值，而应该利用 $N=L/T$ 的关系来赋值。本书中的程序大多数遵循这个原则。用子程序是另一种方法，不过读者不容易取出运行的中间结果，不便于反复温习概念。

从这个例子中还可以比较加大真实数据长度和把数据补零加长的差别。把图 4.5.1 (f)中的幅度频谱重新画在图 4.5.2 (c)中，它是用 *xa* 在 0 到 L=40，采样周期为 0.1s 所得的全部 400 个数据，即

```
xa=exp(-0.1*(1:N)*T);
```

而 *xa*1 是"只取 *xa* 的前 $N2$=100 个数据，而后面 N-$N2$ 个数据补以零"的数组：

```
xa1=[exp(-0.1*[1:N2] *T),zeros(1,N-N2)];
```

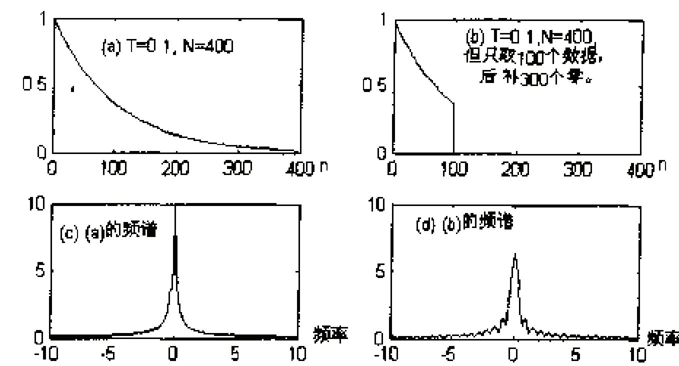


图 4.5.2 数据截断和补零对频谱计算的影响

波形 *xa* 和 *xa*1 分别画在图 4.5.2(a)和(b)中。频谱 *xa* 和 *xa*1 分别画在图 4.5.2(c)和(d)中，从中可以看到波动现象。也许会问，这里所用的 *xa* 和图 4.5.1 (d)所用的原始条件相同，为什么这里出现了波动而图 4.5.1 (d)却没有波动？原因就在于它的长度短，$N*T=L$=10s，其频率为 2π/10=0.628。因为它的分辨率差，所以看不到波动。现在在序列的尾部补上 300 个零，L 变成了 40s，产生的幅频响应就如图 4.5.2 (d)，出现了波动。用准确的理论分析来衡量，把(4.5.2)式的积分上限换成 L 得

$$X_a(\Omega)=\int_0^{\infty}e^{-\alpha t}e^{-j\Omega t}dt=\frac{-1}{\alpha+j\Omega}e^{-(\alpha+j\Omega)t}\Big|_{t=0}^{L}=\frac{1}{j\Omega+\alpha}(1-e^{-(\alpha+j\Omega)L}) \quad (4.5.4)$$

用 4.5.4 式画出的幅频特性也是具有波动特性的。这可以作为用 MATLAB 计算的一个习题。引起波动特性的原因是截断效应，从图 4.5.2 (a)和(b)的比较可以看得很清楚。

因此得出结论，序列补零可以得到高的分辨率，经 FFT 得到很密的频谱，但频谱的值不见得精确，因为它丢掉了许多信息。在实用中，如果可以得到 $e^{-0.1t}$ 的更多数据时就不应该采用在序列尾部补零的方法，因为此时补零可能给出频谱中不正确的细节。

从这个例子可以看出，既然时域截断是计算中人为引入的，频域波动也就是人为引入的。所以首先要解决的是原始时域数据改善的问题，而不是去补零来提高分辨率的问题。

例 4.5.2：用 FFT 计算下列连续时间信号的频谱

$$x_a(t)=e^{-0.01t}\cos 2t+2e^{-0.015t}\sin 2.1t \qquad t\geqslant 0 \quad (4.5.5)$$

在 $t<0$ 时 $x_a(t)=0$。

解：因为信号是用解析形式给出的，用任意长的时间的记录数据都能反映其有效的频率分量。任选 T_1=0.6 及 N_1=256，根据情况逐次再用 T=0.15 和 N=256 及 N=1024 重新计算，最后编出下列程序 hc452，它的核心语句与 hc451 相同。

```
T0=[0.6,0.15,0.15,0.15];          % 各次计算拟采用的T,编成向量T0
N0=[256,256,256,2048];            % 各次计算拟采用的N,编成向量N0
for r=1:4
```

```
    T=T0(r);N=N0(r); n=0:N-1;                    % 根据计算顺序选用T及N
    D=2*pi/(N*T);                                % 频率分辨率D=2*pi/(N*T)
    x=exp(-0.01*n*T).*cos(2*n*T)+2*exp(-0.015*n*T).*cos(2.1*n*T); %序列x
    Xa=T*fftshift(fft(x));                       % 求x的FFT，移到对称位置
  [r,Xa(1)]                                      % 检查各次循环所得奈奎斯特频率处的幅特性
    k=floor(-(N-1)/2:(N-1)/2);                   % 奈奎斯特频率区间下标向量;
    subplot(2,2,r),plot(k*D,abs(Xa))             % 画幅频特性
  end
```

运行此程序可得出图 4.5.3 中的四组曲线。其中图(c)和(d)两个子图放大了横坐标。改变横坐标的语句可以不放在程序内，在实际执行完程序，画完图之后，仍然可以在命令窗中键入 axis ([xmin,xmax,ymin,ymax]) 语句来加以修改。在本例中，就是键入 subplot(2,2,3), axis([1,3,0,60])和 subplot(2,2,4), axis([1,3,0,60])这两条语句而完成图 4.5.3 所示的频谱中下面的两个子图。

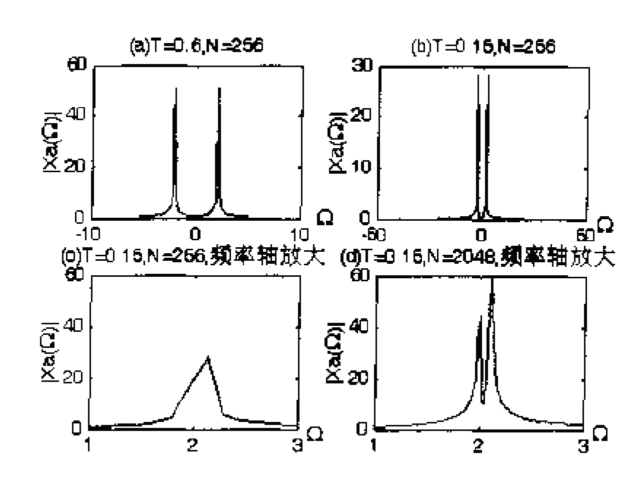


图 4.5.3 具有两个很靠近的谐振分量的信号的频谱

实际上这个程序中的参数是多次试凑才定的，试凑的过程如下。第一次按 T=0.6, N=256 计算后，发现在奈奎斯特频率边界 $\pi/0.6$ 上，Xa(1)=0.8766，而 Xa 的峰值约为 50，超过了 1%。因此要减小 T，当第二次计算中，把 T 取为 0.15,保持 N 不变，得到的 Xa(1)= 0.1779，说明频率混叠已低于 1%。由此得出结论，对 T=0.15，频率混叠实际上可以忽略。此后将用 T=0.15 进行计算，只改变记录的长度 L，因为记录长度直接影响的是频率分辨率，$D=2\pi/L$。要把两个频率很接近的分量的频谱区别开来，就必须有足够大的 L。在 T 一定的情况下，也可以等价地靠改变 N 来调整。

图 4.5.3 (c)把横坐标放大，目的是看信号中的两个谐振峰的情况，可以看出，幅频特性实际上变成了一个峰，根本无法分辨。分辨率低，说明数据长度不够。于是采用了 N=1024 画出幅频特性，也把横坐标放大。两个峰是分得出了，也大体能够看出其幅度的不同。究竟 N=1024 是不是够大了？就必须把它与 N 取得更大的曲线来比较。取 N=2048,，得到图 4.5.3 (d)确实比 N=1024 好，再比较 N=2048 和 N=4096 的结果，它们几乎没有差别，因此就停止计算，总之在两个相继产生的结果相近时就可以停止计算。

同样，也可以用解析解来校验，同上例一样，这里就不重复了。因为在实际中测得的信号，无法知道它的解析解，所以这不是必须要做的工作。主要是要靠相邻两次计算结果的对比来判断是否接近了最后的准确答案。

例 4.5.3： 计算下列信号的频谱。

$$h_a(t)=\begin{cases}\dfrac{\sin 2t}{\pi t} & |t|\leqslant a\\ 0 & |t|>a\end{cases}$$

解设 a=5，这个信号是有限时间信号，它是实的和偶的，因此它的频谱也一定是实的和偶的。当用它的采样序列来计算它的频谱时，很希望它的样本序列是偶对称的，这样由 FFT 算出的频谱也将是实和偶的。为了做到这点，把时间样本数选成奇数，并且要按循环对称的概念排列在 0 到 N-1 的主值下标区间内。

先任选 T=1，则在$|t|\leqslant 5$ 的 $h(nT)$中总共有 11 个样本点，编出如下程序 hc453。

```
T=1;M=5/T;N=2*M+1;n=1:M;                    % 原始数据
D=2*pi/(N*T);                               % 频率分辨率
hp=sin(2*n*T)./(pi*n*T);                    % 生成正序列
h=[2/pi hp fliplr(hp)];                     % 构成循环对称（偶）序列
H=T*fftshift(fft(h));H(1)                   % 求h(n)的FFT，移到对称位置
k=floor(-(N-1)/2:(N-1)/2);                  % 特频率下标向量
subplot(1,2,1),plot(k*D,H)                  % 绘图
```

从 n=1～5 的五个样本在 hp 中产生，在 $t=0$ 处的 $\text{hp}(t)=\left.\dfrac{\sin 2t}{\pi t}\right|_{t=0}=\dfrac{\pi}{2}$，它可以由罗必达法则或偏置 t=eps 求出。由于 ha(t)是偶函数，在 n<0 处的五个时间样本是 hp 的左右折叠值。FFT 的输入必须在 n=0 到 N-1 的主值区间内，因此必须对上面 11 个样本作循环移位。程序 hc454 中的第四行 h 就是这个输入，程序产生了图 4.5.4(a)图的曲线。它在奈奎斯特边界处的值为 0.0074。略低于 1%。取 T=0.5 和 N=21 时，在奈奎斯特边界$\pm\pi/0.5$=6.28 附近只有 0.0005。可见用 T=0.5 采样时的频率混叠可以忽略，后面的计算都将取 T=0.5。

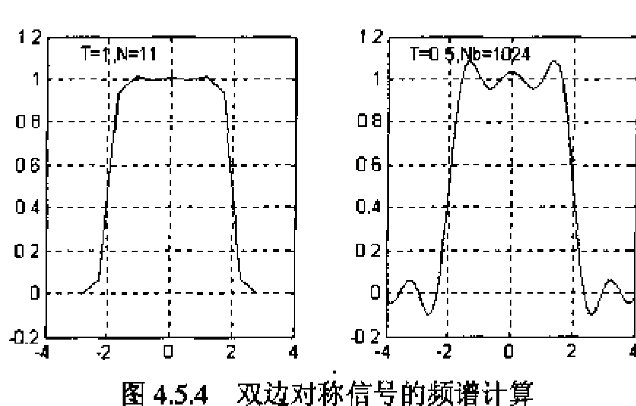


图 4.5.4　双边对称信号的频谱计算

图 4.5.4(a)图频谱的分辨率是很差的，为 $D=2\pi/11=0.57$。因为要计算由$|t|\leqslant 5$ 截断了的 h(t)，提高分辨率的惟一方法是补零。如果采用 1024 点的 FFT，则

```
% 程序hc453 改为hc453a
T=0.5; a=5; M=a/T; N=2*M+1; Nb=1024;        % 原始数据
D=2*pi/(Nb*T); n=1:M;                       % 频率分辨率
hp=sin(2*n*T)./( pi*n*T );                  % 生成正序列
h=[2/pi hp zeros(1, Nb-N) fliplr (hp)];     % 构成循环对称（偶）序列
H=T*fftshift(fft (h)); H(1)                 % 求h(n)的FFT，移到对称位置
k=floor(-(Nb-1)/2:(Nb-1)/2);                % 奈奎斯特频率下标向量
subplot(1, 2, 2), plot(k*D, H)              % 绘图
```

注意对于 T=0.5，只有 21 个数据点，因此必须在序列 h 中引入 1024−21=1003 个零。这些零必须按循环对称的规则来补，而不能补在尾部。这样程序将产生图 4.5.4(b)图的频率响应。这个例子说明用 FFT 确实也能计算连续时间信号的频谱。

4.5.2　连续周期信号的频谱计算

如果一个连续时间信号是绝对可积的，用上节中的步骤计算出的频谱将收敛于真实的频谱。从理论上说，如果一个连续时间信号是周期性的，它必定有无限的长度，计算出的频谱将不会收敛。然而，用计算机计算时，序列总是有限长的。事实上不会遇到真正的周期信号，也不会出现不收敛的问题。我们遇到的都是经过截断了的周期信号。从算法上看，完全可以用上节中的方法和程序，只是在计算结果上，它会有向有限个频点上集中，并成为脉冲的趋势。从下面的例子中看到这一点。

例 4.5.4：考虑定义在全部 t 上的 $x_a(t)=\cos 5t$，它的理论频谱是 $X_a(\Omega)=\pi[\delta(\Omega-5)+\delta(\Omega+5)]$

它包含了权重为 π 的位于 $\Omega=\pm 5$ 上的两个冲击。在计算机计算中，正余弦函数必须截断为有限长度 L。可以采用长度为 L 的矩形窗 $w_L(t)$ 与 $x_a(t)$ 相乘的方法得到截断后的正余弦序列。不断增加 L 就使得序列向真正的周期序列靠近，但永远无法实现周期序列。

信号 $x_a(t)=\cos 5t$ 的带宽限制于 5rad / s，纯理论地看，只要采样周期小于 $\pi/5=0.6283\,\text{s}$，就不会发生频率混叠。然而如果把 cos 5t 截断为长 L 的信号，则它的频谱就不再是有限带宽了，所以必须采用更小的采样周期，假如任选 $T=0.1$，并选 $N=50$，得到 $L=TN=5$。在 MATLAB 中生成截断余弦信号可以有多种方法。可以把余弦函数与矩形窗序列 $R_N(n)$ 点乘，也可以简单地设定一个有限长时间数组 $n=0{:}N-1$，求 $x_a=\cos(\Omega nT)$。参照前面的例子，可以列出程序 hc454 语句如下。

```
N=input('N= '); T=0.1; n=1:N;              % 原始数据
D=2*pi/(N*T);                              % 频率分辨率
xa=sin(5*n*T);                             % 生成有限长的正弦序列
Xa=T*fftshift (fft (xa) ); Xa(1)           % 求h(n)的FFT，移到对称位置
k=floor (-(N-1)/2:(N-1)/2);                % 奈奎斯特频率下标向量
plot(k*D,abs(Xa))                          % 绘图
```

这里把 N 作为用户自选的参数。选 N=50, 100, 500 及 628 所得的计算结果画在图 4.5.5 中，在±5 处都出现了两个尖峰，这是基本的频谱。众所周知，截断将引起混叠和波动。图 4.5.5（a）中只看到混叠而没有波动，图 4.5.5（c）画出了 $T=0.1$，$N=500$ 时得出的频谱。尽管它的频率分辨率达到了 $D=2\pi/50=0.126$，仍看不到波动。实际上当把 N 加倍，两个尖峰的高度也会加倍，并变得更窄，更难见到波动。应该说这是一种好事，因为图 4.5.5（b）和（c）愈来愈像两个脉冲。看来要表现出波动的惟一方法是补零。然而，如果本来能得到更多的信号数据，就没有补零的理由。结论是：如果用 FFT 计算出的幅频谱，包含了窄的尖峰，而 N 每加一倍，它也大体上加倍，这就说明这个连续时间信号包含一个周期分量。这里"大体上"，是因为尖峰窄了以后，很难准确采样在峰值处，所以图上显示的峰值，并非真正的最大值。从理论上说，只有长度 L 恰好是周期的整数倍时，可以得到准确的峰值。

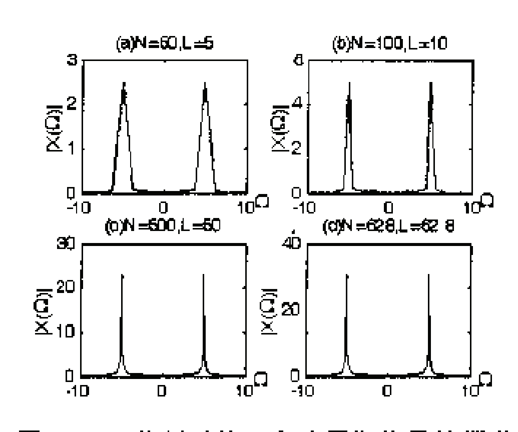
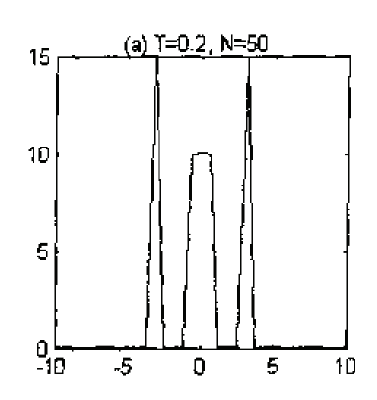


图 4.5.5　截断了的正余弦周期信号的频谱

例 4.5.5：求下列周期信号的频谱

$$x_a(t)=-1+2\sin 0.2\pi t-3\cos \pi t$$

解：它的最高频率是 $\Omega_m=\pi\,\text{rad/s}$，因此采样周期 T 必须小于 $\pi/\Omega_m=1$。任选 $T=0.2$，在图 4.5.6（a）中画出 $N=50$ 时的幅频特性，而在图 4.5.6 (b)中用 $N=1000$。前者由于分辨率太低而只看到三个尖峰，后者则显示出分别位于 $\Omega=0$，±0.628 和±3.14 rad / s 处的五个尖峰，当把 N 加倍，其高度也大体上加倍，所以可以得知这个信号包含了一个直流分量和频率分别为 0.628 和 3.14 的两个正弦分量。



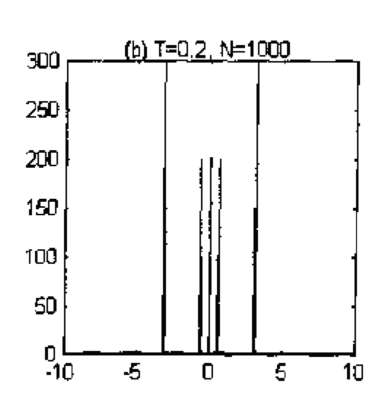


图 4.5.6　含直流及多个分量的正余弦周期信号的频谱

从图 4.5.6 (b)中也可大致看出，在 $\omega=0$，±0.628 和±3.14 的三个高度之比为 2:2:3，由于信号的直流分量是它的零阶傅里叶级数的一半，可见原信号的三个频率分量幅度之比为 1:2:3，所以 FFT 对周期信号做的频谱计算不仅可以确定周期分量的频率，也可以大体确定它的振幅。

由于用傅里叶变换计算连续时间周期信号时，必须截断信号，计算将遇到截断效应。因此对于无限延长的周期信号，如果能够准确的判断其周期，则比较简单的方法是只取其一个周期来计算它的频率分量。这样所有频率分量的幅度可以求得很精确。当然很多实验中得到的信号，很难确定它的周期性，那就只能当非周期信号来分析了。

4.5.3　循环计算中对应频点的确定方法

在例 4.5.1 中，当研究频率混叠时，采用了奈奎斯特频率处的幅度作为定量的依据，比如规定它小于 1%作为混叠可以忽略的准则。但是对于整个频谱曲线在相邻两次计算之间的误差，却只是用目测曲线形状的方法，缺乏定量的数据。其实这个数据也很容易规定，那就是在全频段上求两次计算出的频谱幅度之差，找到全频段上的误差最大值，与全频段上的频谱幅度最大值相比。如果低于 1%或某一个规定值，作为停止计算的准则。在例 4.3.4 的程序中，已经用了这个指标来控制 while 循环的结束。

编这类程序的难点在于，当相邻两次计算中的 N, T 或 L 发生变化时，如何找到可以对应比较其幅度 $|X(\Omega_k)|$的频点 Ω_k? 特别是当两次计算中的点数 N 发生变化，k 也变化了，这个对应关系应怎样建立?

其实只要明确，必须在同样的模拟频点上来对不同的计算频谱进行比较，这个问题就不会搞乱。频率 Ω_k 与采样周期 T，FFT 点数 N 和数据长度 $L=T\times N$ 之间的关系为

$$\Omega_k=\frac{2\pi k}{NT}=kD \qquad k=0,1,\cdots,N-1 \tag{4.5.6 a}$$

假如第一次计算时，用的原始数据是 T_1, N_1 和($L_1=T_1N_1$)；第二次计算时，用的原始数据是 T_2, N_2 和($L_2=T_2N_2$)，两者的 k 也不同。有

$$\Omega_{k1}=\frac{2\pi k_1}{N_1T_1}=k_1D_1 \qquad k_1=0,1,\cdots,N_1-1 \tag{4.5.6 b}$$

$$\Omega_{k2}=\frac{2\pi k_2}{N_2T_2}=k_2D_2 \qquad k_2=0,1,\cdots,N_2-1 \tag{4.5.6 c}$$

频谱样本必须在满足 $\Omega_{k1}=\Omega_{k2}$ 的点上进行比较，将 (4.5.6 a 和 b)式代入此等式，得到

$$k_2=\frac{N_2T_2}{N_1T_1}k_1=\frac{L_2}{L_1}k_1 \tag{4.5.7}$$

只有满足(4.5.7)式的整数 k_1 和 k_2，才是可以进行频谱比较的对应点。如果要得到更形象的结果，可以用由程序 hc456 生成的图 4.5.7 来观察。虽然 k 不同，但它们之间满足整倍数，所以对应频点还是容易找到的。

程序 hc456 采用的核心语句如下：

```
T=?;N=?; L=T*N;D=2*pi/L;
k=floor(-(N-1)/2:(N-1)/2);                      % 设定正负对称频率下标范围
subplot(4,1,r),plot(k*D,0.1*ones(1,N),'.')      % 画出频点集
set(gca,'Xtick',[-pi/T,pi/T]),grid on,          % 标注奈奎斯特频率范围
```

由它生成的图 4.5.7 各个子图只是给定的 T 和 N 不同。假如以第一个子图作为标准情况，T=0.1，N=16，信号长度 L=1.6，那么奈奎斯特频率范围为 $[-\pi/T,\pi/T)=[-31.4,31.4)$[rad / s]，在此范围内等分为 16 份，频率间隔（分辨率）D 为 3.93[rad / s]。下面三个子图是在 T，N，L 三个参数中保持一个不变，而同步地改变另两个所得的结果。

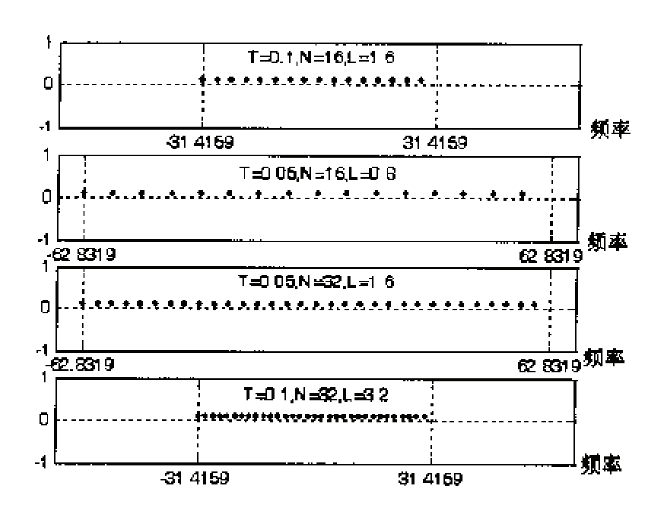


图 4.5.7　改变 T，N 和 L 时，由 FFT 计算的不同频点分布

把 T 减小一半，N 保持不变，L 也减小一半，得到的是第二子图。其主要特点是奈奎斯特频率范围扩大了一倍，成为 $[-\pi/T,\pi/T)=[-62.8,62.8)$(rad / s)。在此范围内仍分为 16 份，所以频率间隔 D 也大了一倍(注意频率间隔与信号长度 L 成反比)，成为 7.86(rad / s)。如果要把它与标准情况进行频谱的比较，只有 8 个点的频谱值可以相减，即 k_1=[-8：2：6]的点集和 k2=[-4：3]的点集对应地比较。

把 T 减小一半，同时把 N 加大一倍，因此 L 保持不变，得到的是第三子图。其奈奎斯特频率范围扩大了一倍，在此范围内分的份数成为 32，所以频率间隔 D 与原来相同，仍为 3.93 [rad / s]。如果要把它与标准情况进行频谱的比较，就有 16 个点的频谱值可以相减，即全部的 k_1 点集和 k_2 中数值与 k_1 相等的部分点集[-8:7]可以对应地比较。k_2 中另一部分数值较大的点集在标准情况中没有对应值，只能舍弃不用。

让 T 保持标准情况，而把 N 加大一倍，此时 L 也加大一倍，得到的是第四子图。其奈奎斯特频率范围与子图一相同，但在此范围内分的份数成为 32，所以频率间隔 D 缩小了一半，变为 1.97 (rad / s)。如果要把它与标准情况进行频谱的比较，也有 16 个点的频谱值可以相减，即全部的 k_1 点集和 k_2 中二抽一的部分点集[-16:2:14]可以对应地比较。k_2 中另一部分点集在标准情况中没有对应值，只能舍弃不用。

所以要编制自动控制循环结束程序时，一是要把前一次循环中频谱数值保存起来；前一次 T 和 N 也要保存。在下一次计算中获得一个新的频谱后，必须先判断前后两个频谱中的哪些点可以对应，然后再计算两者在这些频点上之差的最大绝对值，并按照规定的判别条件决定何时停止计算。

4.6　从频谱计算序列信号（IDFT）

在前几节中，讨论了用 FFT 计算序列和连续信号的频谱，现在来讨论相反的问题，就是从频谱来计算序列和连续信号的问题。为了使头绪更清晰，要把连续信号频谱和序列频谱加以区分，并用表 4.6.1 把正反变换的计算路线做了归纳。表中的第一行把由连续信号求连续频谱的正变换过程作了归纳，就是“采样→FFT→乘 T”三个步骤，变量则经历“连续信号→时间序列→序列频谱→连续信号频谱”四种状态。比如给定时间序列，要求它的序列频谱，那就只要 FFT 一个步骤。本节讨论由连续信号频谱求连续信号的问题，要把步骤反过来，就是

第二行中“除 T→IFFT→求包络”三个步骤，四个变量状态也反过来。比如给出序列频谱求时间序列，那就涉及“IFFT”一个步骤。

变换误差的判断方法也列于表中。正变换的误差为频率混叠，反变换的误差为时间混叠，靠正确选择 T，N 和 L 来控制误差。为了弄清时间混叠的问题，先介绍频率域采样定理。

表 4.6.1　由连续信号求频谱及由连续频谱反求信号的变换过程表

	时间域			正反傅里叶变换	频率域		
变换方向	变量	方法	变量	计算方法 (注意混叠误差的判断)	变量	方法	变量
正变换 (从左向右)	连续信号 $xa(t)$	$x=xa(nT)$ (采样)	离散序列 $x(n)$	X=fft(x), (在折叠频率处的$\|X\|\approx 0$)	序列频谱 $X(k)$	$Xa=TX$ (乘 T)	连续信号频谱 $Xa(\Omega)$; 样本值 $Xa(kD)$; $(D=2\pi/N)$
反变换 (从右向左)		$xa=x(n)$ 取包络		x=ifft(X), (在 $x(n)$ 中有 $x\approx 0$ 的区域)		$X=Xa/T$ (采样并除以 T)	

4.6.1　频率域采样定理

假如已经知道一个序列的频谱 $X(\mathrm{j}\omega)$，要求出它的时间序列 $x(n)$，可以用以下两种方法。

（1）IDTFT 法

$$x(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}X(\mathrm{j}\omega)\mathrm{e}^{\mathrm{j}\omega n}\mathrm{d}\omega \qquad n=0,1,\cdots,N-1 \tag{4.6.1}$$

要完成这个积分，需要知道每一个 ω 处的频谱信息。如果把积分区间等间隔地分为 N 份，每两个相邻频率的间隔为 $\mathrm{d}\omega=2\pi/N$，设频率的序号为 k=0, 1, …, N-1。则这 k 个频点的值为 $\omega_k=k\cdot 2\pi/N$，在这些频点上的频谱值已知为 $Y(k)=X(\mathrm{j}\omega_k)$。如果近似用求和号代替积分，则等价于下一种方法。

（2）IDFT 法

根据（4.1.4）的定义，把 $Y(k)$看做周期频谱 $\tilde{X}(k)$ 在主值区间的值，得到：

$$y(n)=\frac{1}{N}\sum_{k=0}^{N-1}Y(k)\mathrm{e}^{\mathrm{j}kn(2\pi/N)} \quad n=0,1,\cdots,N-1 \tag{4.6.2}$$

显然，$y(n)$ 是 $x(n)$ 的近似，现在讨论它们之间的数学关系和两者的误差估计。 分析的思路是把 IDFT 式中右端的 $X(k)$ 通过 DTFT 式用 $x(n)$表示，这样就把 $y(n)$ 和 $x(n)$ 联系起来了。因为

$$Y(k)=X(\mathrm{j}\omega_k)=\sum_{n-\infty}^{\infty}x(n)\mathrm{e}^{-\mathrm{j}\omega_k n}=\sum_{n-\infty}^{\infty}x(n)\mathrm{e}^{-\mathrm{j}2\pi kn/N} \quad k=1,2,\ldots,N-1 \tag{4.6.3}$$

注意到 $\left.\mathrm{e}^{-\mathrm{j}2\pi kn/N}\right|_{n=N}=1$，可以把 x 分成 r 段（$r\to\infty$），每段的长度为 N，段内的下标设为 $n_1=0,1,\cdots,N-1$，则

$$x(n)=x(n_1+rN) \qquad r=0,\ \pm1,\ \pm2,\ \cdots \tag{4.6.4}$$

将（4.6.4）式代入（4.6.3）式，交换求和次序，得到

$$Y(k)=\sum_{r-\infty}^{\infty}\sum_{n_1=0}^{N-1}x(n_1+rN)\mathrm{e}^{-\mathrm{j}2\pi kn_1/N}=\sum_{n_1=0}^{N-1}\underbrace{\sum_{r=-\infty}^{\infty}x(n_1+rN)}_{x_1(n_1)}\mathrm{e}^{-\mathrm{j}2\pi kn_1/N}$$

于是得到：

$$Y(k)=\sum_{n_1=0}^{N-1}x_1(n_1)\mathrm{e}^{-\mathrm{j}2\pi kn_1/N} \tag{4.6.5}$$

其中：

$$x_1(n_1)=\sum_{r=-\infty}^{\infty}x(n_1+rN)\cdot R_N(n) \tag{4.6.6}$$

它的意义是把整个 x 序列中分成的长为 N 的各段，都叠加到 $0\sim N-1$ 的主值区间上。相当于把无限长的横坐标绕到一个周长为 N 的圆柱体上，把处于圆柱体上同样位置 n_1 的所有 x 值都加在一起。这些 x 有无数个，因为在 $n_1+N, n_1-N, n_1+2N,\ldots$， n_1+rN 处的 x，现在都处在位置 n_1 上了。

可以把(4.6.5)式代入(4.6.2)式继续推导，但可以用更直观的方法。仔细观察 (4.6.2)和(4.6.5)两式，不难看出，它们恰好是一对 DFT 和 IDFT 对偶，把其中的 n_1 换成 n（只考察其主值范围 $0\sim N-1$），显然 $y(n)$就等于$x_1(n)$。也就是说

$$y(n)=x_1(n)=\sum_{r=-\infty}^{\infty}x(n+rN) \qquad 0\leqslant n\leqslant N-1 \tag{4.6.7}$$

由此得出结论：一频谱的采样后经 IDFT 生成的（必然是周期的）序列$\tilde{x}(n)$，等于该频谱经 IDTFT 生成的序列 $x(n)$ 按长度 N 做循环叠加形成的主值序列。(4.6.7)式称为频率域采样定律。它说明，当频率采样点数 N 太小时，可能引起时间域混叠或泄漏，所以和时间域采样引起的频率域混叠或泄漏是对偶关系。当不产生时间域混叠时，对同一频谱求得的 IDTFT 和 IDFT 相等，$x_1(n)=x(n)$，因此可以由 IFFT 来计算和恢复原有的时间序列。当发生时间域混叠时，IDTFT 将不等于 IDFT，由 IFFT 来计算和恢复的时间序列就会有误差，甚至根本不对。

结论是：如果在$\left[0,2\pi\right)$区间内给出了一个频谱，而知道该频谱是由一个有限时间序列产生的，若频率样本数 N 取得比该时间序列的长度大，那么用 IFFT 方法求出的这 N 点 IDFT 就等于 IDTFT 中的包括全部非零点的 N 点序列，IDTFT(它是无限长)中的其余点应为零。由于 IDFT 是定义在 $0{:}N-1$ 的循环下标上的，而 IDTFT 则是定义在线性下标上的，两个下标之间的起始位置会差一个常数 n_0。因此两者的关系应该如下

$$\mathrm{IDTFT}=\begin{cases}\mathrm{IDFT} & n_0\leqslant n\leqslant n_0+N-1\\ 0 & n<n_0\ \text{以及}\ n>n_0+N-1\end{cases} \tag{4.6.8}$$

如果 N 比该时间序列的长度小或该时间序列是无限序列，则会产生时间混叠。不过一无限序列若是绝对可加的，则该序列在 $n\to\pm\infty$时将趋于 0。这时如果把 N 选得足够大，则时间混叠可以变得很小。也可以用相邻两次不同 N 下的计算结果之差小于规定标准来判断。

4.6.2　从频谱计算离散时间序列

本节讨论如何从 X（ω）的 N 个频率样本用 IFFT 来计算 N 点时间序列。为了利用 IFFT 函数，输入的频率样本点必须合乎 IFFT 的要求。这里包括以下几个方面：

（1）给 IFFT 输入的 N 个频率样本应该是数字频率 $\omega=\Omega T$,它的最大值不得超过 2π。通常给定的频谱是以模拟频率 Ω 为横坐标，并同时给出其采样周期 T。如果没给出 T, 它不能像时间序列那样任选，假如频谱的最大频率为 Ωm，采样周期它必须满足：

$$\frac{1}{T}=F_s\geqslant\frac{2\Omega_m}{2\pi},\ \rightarrow\ T\leqslant\frac{\pi}{\Omega_m} \tag{4.6.9}$$

式中，F_s 为采样频率（单位 Hz）。有了适当的 T 或 F_s，才能把给定的频谱 X（Ω）映射到数字频率坐标上来，成为 X（ω）。N 个频点应恰好把 2π 均分，频率分辨率为

$$\Delta\omega = \frac{2\pi}{N} \quad 或 \quad D = \frac{\Delta\omega}{T} = \frac{2\pi}{NT} \tag{4.6.10}$$

所以，输入频谱向量的频率分辨度为 $\omega=2k\pi/N$ 或 $\Omega=kD$ 处。为了区分数字频率与模拟频率，在程序中用 Omega 和 w 两个符号，但往往也有疏漏和混用的情况，所以还是要靠读者的理解。各频点的频率就写成 $k*D$ 或 k*dw，其中频率下标 k 可以是单方向取为 k=0:N−1；也可以取正负频率，此时 k 就取正负双向。

（2）前面指出，通常序列的 DTFT 是用对称于原点的正负频率表示，频率下标取为

$$k=\text{floor}(-(N-1)/2:(N-1)/2)$$

而 IFFT 的输入必须从 $k=0$ 到 N−1，因此，频率样本点 $X(2k\pi/NT)$ 中 $k<0$ 的部分（用 k_n 表示）必须平移到 $k\geqslant 0$ 部分（用 kp 表示）的右面。其中 $k_n=\text{floor}(-(N-1)/2:-1/2)$，而 $k_p=\text{floor}(0:(N-1)/2)$，即把 k=0 放在 kp 中。

（3）若待求的时间序列是实序列，则输入频谱 $X(\Omega)$应具有对称性。$X(k)$应具有循环共轭对称性。所以把 k 作上述移动不影响其对称性。计算出的输出序列应该是实的，但由于运算误差，也可能有微小的虚部。如果计算出的输出序列中含有数值可观的虚部，则应检查输入频谱的对称性。

经过这样生成的 X，才合乎要求。用 IFFT 求出的 N 个数据点是 $\tilde{x}(n)$ 在 $n=0:N-1$ 主值区内的值 $x_1(n)$。然后以上节的理论推导为基础，具体地分析这个 IDFT 值 $x_1(n)$中的时间混叠。由上节得知，频率采样会引入时间混叠。如果时间混叠不能忽略，那样算出的结果是无用的。检查时间混叠的方法如下：如果算出 $x_1(n)$对所有的 n 都明显地不等于零，说明时间混叠相当大，必须另选一个大些的 N。当算出的 $x_1(n)$在 n 的某个范围内实际上等于零了，即 IDFT 存在着(4.6.8)式中含零的部分，就可认可这个 N。下面的例子将说明这个过程。

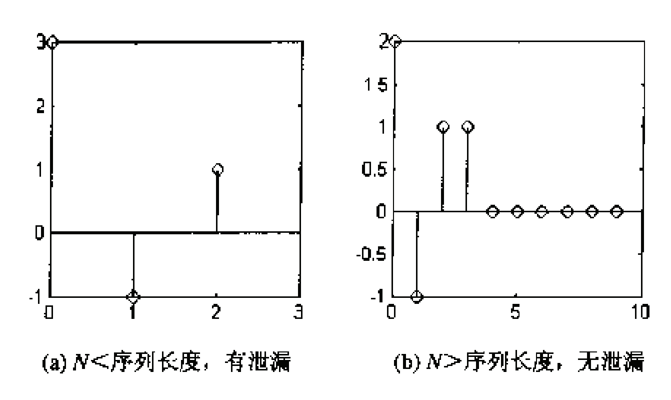
例 4.6.1：设某时间序列的频谱为 $X(\mathrm{j}\omega)=\sum_{-\infty}^{\infty}x[n]\mathrm{e}^{-\mathrm{j}\omega n}=2-\mathrm{e}^{-\mathrm{j}\omega}+\mathrm{e}^{-\mathrm{j}2\omega}+\mathrm{e}^{-\mathrm{j}3\omega}$。用 FFT 数值计算的方法求出它的时间序列。

解：这个频谱实际上是例 3.2.1 中求出的。如果用这样明显的解析式给出频谱，它对应的时间序列一眼就可以看出为 $x(n)$=[2, −1, 1, 1]，根本无需计算。但我们又没有办法用数值方法给出全频段的频谱。所以就请读者用数值分析的角度来看待这个已知条件。

先取 N=3，因为题中用的是数字频率而没有涉及 T，这就是默认 T=1，因为要采用 IFFT，它要求的输入频率数组必须是把 2π 等分为 N 份的数字频率，则将下列语句补入程序：

```
T=1;N=3;D=2*pi/N;                              % 频谱序列长度及频率分辨率
kn=floor(-(N-1)/2: -1/2);                      % 负频率下标向量
kp=floor(0:(N-1)/2);                           % 正频率下标向量
w=[kp,kn]*D;                                   % 将负频率移到正频率的右方，形成新的频率排序
X=2-exp(-j*w)+exp(-j*2*w)+exp(-j*3*w);         % 按新的频率排序输入数字频谱
x=ifft(X);                                     % 对循环对称的 X 求 IFFT，得出循环对称的 x
stem(T*[0:N-1],x)                              % 绘图
```

N 取值为 3 和 10 两种情况得出的结果见图 4.6.1。当 N=3 时，因为 N 小于时间序列长度，产生了时间混叠，其标志是序列中没有取零的点或点列，结果不正确；当 N=10＞4 时，结果是正确的，其标志则是在算出的 IDFT 序列 $x(n)$中，包含一段如(4.6.8)式所指的全零的部分，



(a) N<序列长度，有泄漏　　(b) N>序列长度，无泄漏

图 4.6.1　用 IDFT 从频谱求序列时频率样点数 N 对时间泄漏的影响

下面的例子是用模拟频率为输入数据的。

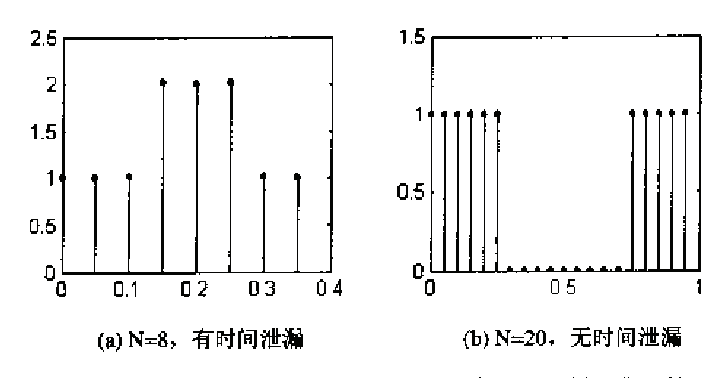
例 4.6.2：给出在频率范围[−62.8, 62.8] rad/s 间的频谱。

$$X(\Omega)=\frac{\sin 0.275\Omega}{\sin 0.025\Omega} \tag{4.6.11}$$

求出它的时间序列。

解：本题给出的是模拟频率 Ω，Ωm=62.8，需将它归一化为数字频率。按(4.6.10)式，取 T≤π/Ωm=0.05s。取 T=0.05，即采样频率为 20Hz 或 40π。如果分成 N 份，模拟频率分辨率（也就是频率增量）D 按（4.6.10）式确定。而各频点的频率按 $k*D$ 即可确定。因为（4.6.11）式中的 $X(\Omega)$的周期为 $N*D$，可以直接取 k=0: N−1 不必采用先取正负频率再移位的方法。在计算 $\sin(0.275k\cdot D)/\sin(0.025k\cdot D)$ 时，为了避免 k=0 处出现 0/0 的不定式，有意把 k 偏移了一个微量 eps。把这些考虑全放到 MATLAB 程序中，分别按 N=8 和 N=20 两种计算，写出程序 hc462。

```
wm=62.83;T=pi/wm;                        % 选择采样周期 T
N0=[8,20];                               % 设定两种 N 值
for r-1:2                                % 做两次循环计算
   N=N0(r);D=2*pi/(N*T);                 % 取 N,并求出模拟频率分辨率
   k=[0:N-1]+eps;                        % 求频率下标，偏移微量
   X=sin(0.275*k*D)./sin(0.025*k*D);     % 算出频谱 X
   x=ifft(X);                            % 求 X 的 IDFT，得出 x(n)
   subplot(1,2,r),stem([0:N-1]*T,x)      % 绘图
end
```



(a) N=8，有时间泄漏　　(b) N=20，无时间泄漏

图 4.6.2　用 IFFT 求时间序列时 IFFT 长度 N 对时间泄漏的影响

程序运行得出的曲线见图 4.6.2。N=8 时得到(a)图，其中的八个数据中没有一个为零，可见存在着频率采样所造成的时间混叠，再取 N=20 进行计算。结果画在 4.6.2(b)图中，算得的序列中都有 11 个非零的数据，其中确有一个范围内算得的 x 都等于零，结果是正确的。

对于这个时间序列进行周期延拓，可知这些非零的数据存在于 n=−5:5，因此有

$$x(nT)=1 \qquad n=-5:5$$

而在其他所有的 n 上 $x=0$，这就是例 3.2.1 中的序列矩形窗。因此，用 IFFT 算出了正确的时间序列，或者说算出了（4.6.11）式的离散时间傅里叶反变换。因此如果取 N=12 或者更大，不管 N 取多少，都没有时间混叠，得到 11 个 1。这也从反方面证明了时间序列尾部补零不影响频谱形状的结论。

如果一个频谱的 IDTFT 具有无限长，则因频率采样而造成的时间混叠将永远存在。然而，如果频谱是有界的和连续的，则它的 IDTFT 必然是绝对可加的。因此在 $|n|\to\infty$ 时，此时间序列必然趋于零。把 N 取得足够大总能得到近似的有限序列。

前面所举的例子中，信号在 n<0 时的值都等于零，即它们是正时间信号。当一个频谱的 IDTFT 是正时间信号，而且频率样本数 N 大于时间序列的长度（从 $n=0$ 算起）则由 IFFT 生成的序列等于原始序列。如果出现了负时间信号，算出的序列形状是对的，但序列的原始起点可能不对。这时要依靠取不同的素数 N 做多次计算，并把它们做周期延拓后，从对比中找到原始起点。因为工程中不会遇到负时间信号，所以本节就不做讨论了。有兴趣的读者可参阅参考文献[8]。

4.6.3　从频谱计算连续时间信号

本节讨论利用 IFFT 来计算连续时间傅里叶反变换的问题，或者说从频谱 $X_a(\Omega)$ 计算 $x_a(t)$。从本质上说，计算机并不能处理连续信号，它只能把连续信号用一个采样周期极小的序列信号来近似，所以上节所有的理论和程序仍然有效。仍然要选择一个非常小的 T 使得主要模拟频谱处于奈奎斯特频率范围之内

$$X_a(\Omega)\approx 0 \qquad 当|\Omega|\geqslant \pi/T \tag{4.6.12}$$

然后在 $[-\pi/T,\pi/T)$ 频率区间内取 N 个频点 Ω_k，并根据(4.4.10)求出其数字频谱 $X(\Omega_k)$，使 $X(\Omega_k):=X_a(\Omega_k)/T$，并对 $X(\Omega_k)$ 做 IDFT，得到一个采样周期为 T 的以 N 为周期的主值序列 $x_1(n)$。假如没有时间混叠，可以求出 $x(n)=x_1(n)$=IDFT[X(Ω_k)]=IDFT[Xa(Ω_k)]/T。现在要从 $x(n)$ 求出对应的连续信号 $x_a(t)$。

$$x_a(t)\big|_{t=nT}\approx x(n)=IDFT[X_a(\Omega_k)/T] \tag{4.6.13}$$

这里我们已经隐含地假设 $x(n)$ 被内插而形成了一个连续时间信号。如果 $X_a(\Omega)$ 的带宽 Ωm 有限，T 又小于 π/Ωm，近似式（4.6.13）就趋向于等式。

设 $x_1(n)$ 是 $X_a(\Omega)$ 的 N 个频率样本的 IDFT，如果对某些 n_0 有

$$x_1(n)=\begin{cases}x(n) & 对\quad n_0\leqslant n\leqslant n_0+N-1\\ 0 & 对\quad n<n_0和n>n_0+N-1\end{cases} \tag{4.6.14}$$

当 $x_1(n)$ 或 $x(n)$ 是时间有限的，并且 N 足够大时，这近似式的误差就可以忽略。把（4.6.13）与（4.6.14）结合起来，得到

$$x_a(t)\big|_{t=nT}\approx x_1(n)=\begin{cases}x(n) & 对\quad n_0T\leqslant t\leqslant(n_0+N-1)T\\ 0 & 对\quad t<n_0T和t>(n_0+N-1)T\end{cases} \tag{4.6.15}$$

这里我们又隐含地假设 x 已经被内插为一个连续时间函数。不过这个近似式永远不会变成等式。因为 $x_a(t)$ 不可能同时为带宽有限和时间有限的。即使如此，仍然可能得到一个由 IFFT 算得的，可以任意接近真实连续时间信号的信号。下面就是它的核心语句。

```
wm=? ;T=0.1*pi/wm;              % 选择采样周期T，尽量取算出的数值的1/10左右
N=? ;D=2*pi/(N*T);              % 频谱序列长度及频率分辨率
kn=floor(-(N-1)/2:-1/2);        % 负频率下标向量
kp=floor(0:(N-1)/2);            % 正频率下标向量
w=[kp,kn]*D;                    % 将负频率移到正频率的右方，形成新的频率排序
Xa=?                            % 按新的频率排序输入给定的频谱
xa=ifft(Xa/T);                  % Xa/T是序列频谱，其IFFT为时间序列x(n),xa(n) =x(n)
plot([0:N-1]*T,x)               % 用plot语句绘图，实现插值得出连续信号xa(t)
```

用这个程序可以得到近似的 $x_a(t)$。当 T 减小而 N 增加时，计算结果就更接近于真实值，这个程序与程序 hc461 的不同只有两个部分，一是把 Xa 要除以 T，因为要把模拟信号频谱变为数字序列频谱；二是用 plot 代替 stem，因为 plot 可以自动进行直线内插，获得连续信号。在程序的应用上，为了弥补直线内插的不足，避免再进行内插编程，在选择 T 时可把它比公式（4.6.9）小一个数量级，相应地要选较大的 N。N 的选择要考虑两个因素：一是（不等于零的）有效频率样本要足够多，最好要超过 100 个，这可保证频率步长不到有效带宽的百分之一，以保证输入频率数据的本身精度高于 1%。列出关系式为

$$D=\frac{2\pi}{NT}\leqslant 0.01\cdot\Omega m \rightarrow \qquad N\geqslant\frac{200\cdot\pi}{\Omega m\cdot T} \tag{4.6.16}$$

二是最终输出模拟信号的长度 $N*T$ 满足命题的要求，太短了无法充分反映模拟信号的变化。下面给出两个例子。

例 4.6.3：考虑图 3.2.5 所示的模拟理想低通滤波器，参数改为 $\omega_c=3$，在 $|\omega|\leqslant\omega_c=3$ 处的幅度为 1，而在 $|\omega_c|\geqslant 3$ 的幅度为零．要求计算它的连续脉冲响应，也就是它的连续傅里叶反变换（ICTFT）。

解：因为频谱的带宽限于 ± 3 rad / s，如果选择 $T<\pi/3=1.06$，那么就不会产生频率混叠。然而，因为我们想由 $x_1(nT)$ 由直线插值得到 $x_a(t)$，而不想用插值公式，T 必须取得小，因此要选得比 1.06 小一个数量级。取 $T=0.1$s，这相当于 10 Hz 或 62.8 rad / s 的采样频率。相应的奈奎斯特频率范围将是 $(-\pi/T,\ \pi/T]=(-31.4,31.4]$。按 $\Omega_m=3$，$T=1$ 代入(4.6.16)式，大体得到 $N=2000$。现选 $N=2048$，则频率分辨率就是 $D=2\pi/(NT)=0.0307$。这样在带宽 $\omega_c=3$ 的有效频段内，大概有 100 个样本。找到它的边界所对应的频率下标 $M=\lfloor\omega_c/D\rfloor$。用 MATLAB 语句为 M=floor(wc/D)=floor(97.8)=97，于是可写出该频谱的样本值为

$$X(k\cdot D)=\begin{cases}1 & 对于 k=0,\pm1,\cdots,\pm M\\ 0 & 其他地方\end{cases}$$

整个计算过程可以用下面的 MATLAB 程序 hc463 归纳起来。

```
% 用IDFT由已知频谱求连续时间傅里叶反变换ICTFT信号
wc=3;Tmax=0.1*pi/wc             % 最大采样周期Tmax，取算出临界数值的1/10
T=input('(T<Tmax)T= ')          % 用户选择并输入T
Nmin=100*2*pi/wc/T              % 用(4.6.16)式求最小样点数目Nmin
N=input('(N>Nmin)N= ');         % 用户选择并输入N
D=2*pi/(N*T); w=[0:N-1]*D;      % 模拟频率的频率分辨率及频点向量
M=floor(wc/D);                  % 有效频率的边界下标（整数）
xa=[ones(1,M+1),zeros(1,N-2*M-1),ones(1,M)];    % 按新的频率排序输入给定的频谱
xa=ifft(Xa/T);                  % Xa/T是序列频谱其IFFT为时间序列（x(n), x(n)=xa(n)）
plot([0:N-1]*T,xa)              % 用plot语句绘图，实现插值，得出连续信号xa(t)
```

运行此程序，得到 Tmax=0.106，按提问键入 $T=0.1$ 后，得到 Nmin=2094。再按提问键入 $N=2048$。最后得到图 4.6.3a 中所示的连续信号。如果观察 xa 的数值，得知 xa(320)= 0.0096。就是说，在 $|t|>32$ s 后，实际上可把它看成零，因此满足了条件(4.6.14)式。若取 $N=4096$，得出由虚线表示的曲线。它比 $N=2048$ 的时间波形长了一倍，但它的起始部分和 $N=2048$ 的曲线完全重合。这个脉冲响应其实只有前半段有用，后半段

是它的镜像，它是由于计算中用了 IDFT，引进了循环对称性所造成的。

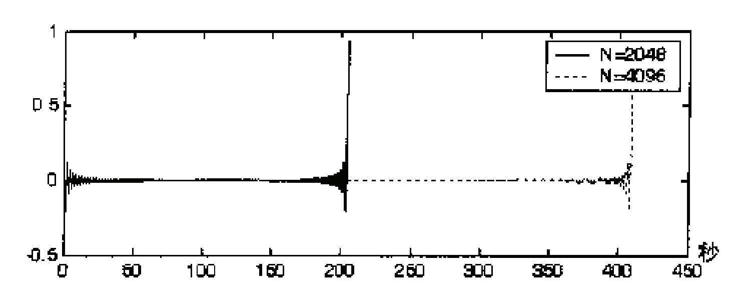


图 4.6.4　理想低通滤波器的脉冲过渡函数，用 IDFT 求 ICTFT

由此得出结论：题中频谱的连续傅里叶反变换就如图 4.6.3 中所示的那样，这个曲线非常接近解析解的结果。上述例子说明，连续时间冲激响应确实能用 IFFT 从它们的频谱求得。

4.7　用 FFT 计算能量

在第 4.4 节中谈到了频率混叠的严重程度可以用能量来评价。本节将讨论用 FFT 计算连续信号的 $x_a(t)$ 能量，根据定义，$x_a(t)$ 的总能量为

$$E=\int_{t=-\infty}^{\infty}|x_a(t)|^2\mathrm{d}t=\frac{1}{2\pi}\int_{\omega=-\infty}^{\infty}|X_a(\omega)|^2\mathrm{d}\omega \tag{4.7.1}$$

因此总能量可以直接在时域中求得，也可以从频域中先算出频谱后再间接地求得。如果算出了频谱，不仅可以算出总能量，而且可以算出任何频率区间内的能量。

为了在时域中直接求出总能量，可以把(4.7.1)式近似为

$$E=\sum_{n=n_1}^{n_2}(x_a(nT))^2T \tag{4.7.2}$$

n_1 和 n_2 决定了计算的时间区间，在这个方程中 $x_a(t)$ 是用图 2.1.1(b)那样的阶梯信号来近似的．并且假定它在 $t<n_1$ 和 $t>n_2$ 处为零。

正如在第 4.3 节中讨论的，数字频谱 $X(k)$可以由 $x_a(t)$ 的样本 $x(n)$，选择恰当的 T 和 N，利用 FFT 算出。设 $X(k)$为由 $x(n)$算得的频谱，则等价的模拟频谱 X_a 由下式得到

$$X_a(k\cdot D)\approx T\cdot X(k) \tag{4.7.3}$$

对于 k=floor(−(N−1)/2): (N−1/2))和 $D=2\pi/NT$ 成立。利用公式（4.7.1）和（4.7.3），总能量可近似为

$$E=\frac{1}{2\pi}\sum_{k=-(N-1)/2}^{(N-1)/2}|TX(k)|^2D=\frac{T}{N}\sum_{k=0}^{N-1}|X(k)|^2 \tag{4.7.4}$$

这是连续信号 $x_a(t)$ 的总能量。一旦知道了 $X(k)$，只要选择适当的 k 的范围，也可以计算任何频率区间的能量，这将在下一个例子中说明。请注意：由于能量只与振幅特性有关，因此它不会受时移的影响，在使用(4.7.4)式时，没有必要在应用 FFT 之前重新排列 $x(n)$。另外，更方便的方法是只利用 $[0,\ 2\pi/T)$ 正半频段，而不必用全频段。因此更没有重新排列的必要了。

例 4.7.1：考虑矩形窗 $x_a(t)=\begin{cases}1 & |t|\leqslant a\\ 0 & |t|>a\end{cases}$

设 $a = 2$，求出其频谱在主瓣中的能量所占总能量的百分比。

解：此窗的总能量可以在时域中直接算出：

$$E = \int_{t=-\infty}^{\infty} |x_a(t)|^2 \mathrm{d}t = \int_{t=-2}^{2} 1^2 \mathrm{d}t = 1 \cdot (2-(-2)) = 4$$

为了求得在各种频率上的能量分析，可只计算它的频谱，虽然它的频谱可以解析地求出，现在用本节中的步骤，再对它作数值计算。先取采样周期为 $T=0.01$，然后求得边界下标 $M=a/T=200$。因而时间序列为

$$x(nT)=1 \qquad -M \leqslant n \leqslant M$$

这是一个长度为 401 的有限序列，现用 FFT 来计算其离散时间傅里叶变换，为了得到较高的频率分辨率，采用 $N=16384$。因为能量只依赖于幅度频谱，所以将不对 FFT 的输入重新排序。可编出下列程序 hc471：

```
% 求矩形窗频谱能量的程序 hc471
T=0.01;N=16384; a=2;              % 输入原始数据
D=2*pi/(N*T);                     % 频率分辨率
M=floor(a/T);                     % 求边界点下标
x=[ones(1,2*M+1)];                % 给出时间序列
X=fft(x,N);                       % 求傅里叶变换 FFT
k1= input('k2= ');                % 求能量的起点频率下标
k2=input('k2= ');                 % 求能量的终点频率下标
k=k1:k2;                          % 求能量频段的下标序列
E=T/N*sum(abs(X(k+1)).^2)         % 求区间能量
```

键入 k1=0 及 k2=16383,计算出总能量为 4.01，这与矩形窗精确解的结果 4 只差了 0.25%。Xd 主瓣的范围从 $w=-\pi/2$ 到 $\pi/2$，我们将计算[0，$\pi/2$]区间内的能量，然后把它加倍。先计算边界下标 k1＝0,k2=floor(0.5π/D)，并按提示输入，就可以求出主瓣能量，如果在程序后面加上下面的语句：

```
E1=2*T/N*sum(abs(X(k+1)).^2)
E1/4.01*100
```

则程序将得到E=4.01, E1=3.7185, 和92.73%。说明在主瓣中的能量为3.7185, 它占了总能量的92.73%。

4.8 小　结

最后要说明的是傅里叶分析的某些局限。一个信号可以在时域表示为 $f(t)$，也可在频域表示为 $F(\omega)$。因为任何一个都可从另一个得到，所以两种描述方法包括了同样的信息量。

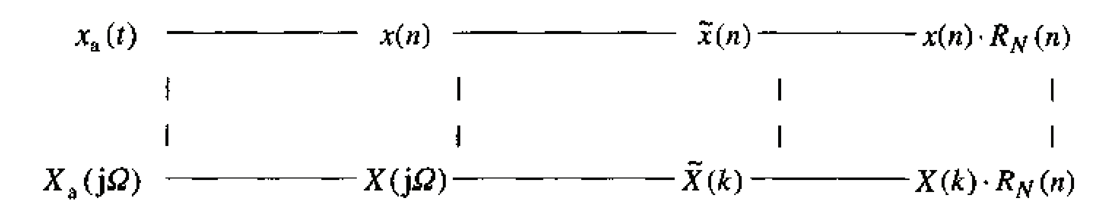
如果能把这两种描述结合起来，就有可能得知信号的更多内涵。不过这是困难的，因为信号的傅里叶变换是从信号的全部持续时间内，加了同样的权重 $|e^{j\omega t}|$ 而发展出来的，时间信号的局部特性在它的傅里叶变换中就成为一个全局的频率特性，反之亦然。例如，在某一个瞬间的信号的不连续性，不管发生在什么地方，在频域中就会使所有的频率分量不为零。因为时移不影响幅频特性，时间位置的信息必然包含在相位谱中，然而相位对计算是相当敏感的。而且会因为计算中的采样而造成严重的混叠，所以要得知时间位置是困难的。在傅里叶分析中要得知频域表述和时域表述之间的相关性是较困难的。

补救这个问题的方法之一是把一个时间信号分成好几段，然后分段计算它们的短时间加窗的傅里叶变换。显然，每一个时间区段愈窄，那它的时间位置就愈准确，不过计算出的变换却愈不精确。一个窄的时间窗会有一个较宽的频率主瓣，形成的频谱要和窗函数的频谱在一个较宽的范围内卷积，因而又变得不精确。比较好的一个办法是利用小波(wavelet)和小波

变换，以便较好地建立信号的时间和频率之间的相关性。

复习思考题

4.1 画出一张图，表明各信号形式和它们的傅里叶变换的形式之间的关系，其基本格式为：



在图上的线条处，画出双向的两个箭头，用文字标明变换的关系式，并注明变换中可能产生的问题或误差。

4.2 FFT 能提高计算效率的基本思想是什么？

4.3 何谓蝶形运算？利用蝶形运算为何可以节省一半乘法次数？

4.4 何谓位倒序?何谓原位计算?这两个概念有何联系？

4.5 是不是 FFT 只能用于数据点数等于 2 的幂的情况?是不是基数分解得愈小愈好？

4.6 在 N=8 的情况下，把求 FFT 的子程序 myditfft 与图 4.2.1 的流程对照起来进行解释。

4.7 用 MATLAB 写出一个基四的 DIT FFT 子程序 m4ditfft.m。考核它的计算速度。

4.8 为什么用 FFT 计算出的频谱，其频率排序是从零到 2π？如何使它变成对零频率对称的形式并作图？用自编 DFT 程序计算时，是否可以自选频率排序的编号？

4.9 如果由 FFT 计算出了频谱曲线上的 N 个样本点，如何求出样本点之间的频率处的频谱？有几种方法？各有什么优缺点？

4.10 为什么提高序列的频谱精度的主要方法是增加样本长度？用补零的方法能否提高频谱的分辨率？能否提高它的精度？

4.11 用计算机分析无限序列频谱时，必须把它截断。截断对频谱分析的结果有哪些不良影响？怎样避免和减少这些影响？

4.12 时域连续信号的频谱及其取样信号的频谱之间的关系是什么？

4.13 什么是频率混叠？它是如何产生的？如何减小混叠？

4.14 对于带限信号，应如何选择采样频率？对于非带限信号，如何选择采样频率？

4.15 为什么在 A/D 变换器的前面通常放置低通预先滤波器？

4.16 把数字信号重构为模拟信号时，要经过哪两个主要步骤？为什么？D/A 变换器是如何实现这两个步骤的？

4.17 低通预先滤波器和 D/A 变换器后的平滑低通滤波器在要求上有什么不同？

4.18 用本章复习思考题 1 的框图，说明通过 FFT 从连续时间信号计算其频谱所经过的主要步骤。并说明其中可能带来误差的环节。

4.19 在选择计算参数时，采样周期 T，样本点数 N 和序列长度 L 分别对频谱特性的哪个指标产生直接的影响？当要提高分析的频谱宽度和提高频谱的分辨率时，各应该首先考虑修改哪个参数？

4.20 用傅里叶反变换求时间函数前，为什么必须先把频率特性数据调整得合格？合格的

主要标准是什么？

4.21 什么是频率域采样定律？什么是时间混叠？为什么在求傅里叶反变换时要考虑时间混叠？时间混叠与频率混叠有什么异同？减小时间混叠的主要措施是什么？

习题

4.1 如果计算机平均每次复数乘法需要 5μs, 每次复数加法需要 1μs,若用普通方法计算 1024 点的 DFT，需要多少时间？用基二的 FFT 计算呢？按这样的速度对信号进行快速卷积实时处理，问信号的最高采样频率是多少？

4.2 如果采用数字信号处理专用的 TMS320 序列芯片进行处理，每次复数乘法需 400 ns,每次复数加法需 100 ns,重算习题 4.1。

4.3 考虑两个序列：（a）$x(0)=1$，$x(1)=-2$，$x(2)=1$；（b）$x(0)=-1$，$x(1)=2$，$x(2)=2$，$x(3)=-1$。如果采样周期 $T=0.5$，试用一个四点 DFT 来计算这两个序列频谱的四个频率样本。

4.4 用 1024 点 FFT 来计算并画出题 4.3 中两个有限序列的连续频谱。

4.5 证明习题 4.3(a)序列中的三点 DTFT 频谱满足。

$$X_a(\omega)=\frac{1}{3}\sum_{m=0}^{2}X(k)\frac{1-e^{-j3(0.5\omega-2\pi k/3)}}{1-e^{-j(0.5\omega-2\pi k/3)}}$$

这个公式可以用来由样本值 $X(k)$求出其连续频谱，把它的结果与习题 4.4 的结果进行比较讨论。对习题 4.3 (b)序列中的四点 DFT 的频谱，这个公式应该做什么改变？

4.6 解析地计算 $x_a(t)=e^{-0.2t}, t\geqslant 0$ 的频谱 $X_a(\Omega)$。又设 $T=0.5$，计算 $x(n)=e^{-0.2nT}$，$n\geqslant 0$ 的频谱 $X(\omega)$，把两者进行比较，讨论是否存在着因采样引起的频率泄漏。若改为 $T=0.05$,其结果如何？

4.7 序列(a) $x(n)=0.95^n$，$n\geqslant 0$; 序列(b) $x(n)=(-0.95)^n$，$n\geqslant 0$。计算它们的 DTFT 并作其频谱曲线。讨论其中哪个是高频信号。

4.8 对模拟信号 $x_a(t)=2\sin(4\pi t)+5\cos(8\pi t)$ 在 $t=0.01n$，$n=0,1,\cdots,N-1$ 上采样，得到 N 点序列 $x(n)$，用 N 点 DFT 得到对 $x_a(t)$ 幅度谱的估计。

(a)从下面值中，选择一个能提供最精确的 $x_a(t)$ 的幅度谱的 N,画出 DFT 幅度谱$|X(k)|$的实部和虚部。① N=40　② N=50　③ N=60

(b)从下面值中，选择一个能提供最小 $x_a(t)$ 的幅度谱泄漏量的 N,画出 DFT 幅度谱$|X(k)|$的实部和虚部。① $N=90$　② $N=95$　③ $N=99$

4.9 求出并画出 $N=8$ 点，基二的 DIF(频率抽选)FFT 算法的信号流图，根据流图，确定下面序列的 DFT。

$$x(n)=5\cos(\pi n/2)\qquad 0\leqslant n\leqslant 7$$

4.10 求出并画出 $N=16$ 点，基四的 DIT FFT 算法的信号流图，根据流图，确定下面序列的 DFT。

$$x(n)=2\sin(\pi n/2)\qquad 0\leqslant n\leqslant 15$$

4.11 设 $x(n)=\cos(\pi n/99)$，$0\leqslant n\leqslant(N-1)$为 N 点序列。选择 $N=4^{\nu}$，求出 $\nu=5,6,\cdots,10$ 的 MATLAB 执行时间。验证这些时间与 $N\log_{\nu}N$ 成正比。

4.12 考虑无限序列 $x(n)=0.9^n$（$n\geqslant 0$），$T=1$。求出最小的 a 使得前 $N=2^a$ 个数据点与无

限序列之间的频谱误差小于峰值的 1%。只在 2mπ / N（m=0:N / 2）的点上比较其幅值。

4.13 同题 4.12，只把前 N 个数据点改为前 $N/2$ 数据点。比较与上题结果的差别。

4.14 同题 4.13，把序列改为 $x(n)=(-0.9)^n$（$n\geqslant0$）。

4.15 考虑连续信号 $x(t)=2\,\mathrm{e}^{-0.5t}$（$t\geqslant0$），求它的频谱的解析式。用 FFT 按下列步骤计算它的频谱：

(a)求出 L，使时间记录长度（0，L）只丢掉信号幅度低于最大幅度的 1%的部分，用这个 L 来求得采样周期 T 以及在 $N=2^a$ 中的最小 a 值，使频率泄漏可以忽略不计，也即用 T 和 N 算出频谱幅度与用 $2T$ 及 $N/2$ 算出的结果差别小于峰值的 1%；

(b)利用在步骤（a）中得到的 T 和 N 求得在 $Nm=2^c$ 中最小的 c 值，使得截断效应可以忽略不计，也即利用 T 和 Nm 求出的频谱与用 T 和 $Nm/2$ 的结果之差小于峰值幅度的 1%。

(c)把步骤（b）中得到的频谱与精确值进行比较，同样比较其相位特性。

4.16 用 FFT 来计算下列信号的频谱。

$$x_a(t)=\mathrm{e}^{-0.01t}\cdot\sin 2t-3\mathrm{e}^{-0.02t}\cdot\cos 5t$$

4.17 用 FFT 来计算列信号的频谱。

$$x_a(t)=\mathrm{e}^{-0.01t}\cdot\sin 3t-3\mathrm{e}^{-0.02t}\cdot\cos 3.05t$$

其最后的时间记录长度与第 4.15 题相同吗？

4.18 一个连续信号 $x_a(t)$ 由频率为 250 Hz, 450 Hz, 1.0 kHz, 2.75 kHz 和 4.05 Hz 的正余弦信号的线性组合。此信号被 1.5 kHz 的采样频率采样后，通过一个截止频率为 750 Hz 的理想滤波器，生成一个连续信号 $y_a(t)$，试问此重构信号 $y_a(t)$ 中所包含的频率分量。

4.19 一个连续信号 $x_a(t)$ 由频率为 F_1, F_2, F_3 和 F_4 的四个正余弦信号的线性组合。此信号被 8 kHz 的采样频率采样后，通过一个截止频率为 3.8 kHz 的理想低通滤波器，生成一个连续信号 $y_a(t)$，此重构信号 $y_a(t)$ 中包含有 450 Hz, 625 Hz 和 950 Hz 的三个频率分量。请问 F_1, F_2, F_3 和 F_4 的可能频率值是多少？答案是否惟一？能否再找出一种解答？

4.20 考虑下列频谱：

$$X(\omega)=\mathrm{e}^{\mathrm{j}\omega}(2\cos\omega-2) \qquad |\omega|\leqslant\pi$$

(a) 在 $k\cdot(2\pi/3)$ 处计算它的三个频率样本的 IDFT。

(b) 计算它在 $k\pi/2$（k=0，1，2，3）处的四个频率样本点的 IDFT。

(c) $X(\omega)$的 IDTFT 是什么？

4.21 对 $X(\omega)=2\mathrm{j}\sin\omega(|\omega|\leqslant\pi)$，重复题 4.20 的内容。

4.22 考虑 $X(\Omega)=\dfrac{2}{(\mathrm{j}\Omega+1)^2+4}$ $(-\infty<\Omega<\infty)$，用 IFFT 计算它的时间函数。

第 5 章 变换域中的离散时间系统

5.1 z 变换

前面讨论的离散傅里叶变换是一种用复指数序列表示离散信号的方法。它通过频率响应函数 $H(e^{j\omega})$ 将系统函数表示在频域中。很明显，这种表示方法用 $H(e^{j\omega})$ 函数简化了正弦稳态响应的计算，对 LT1 系统很方便。此外,任意绝对收敛的序列 $x(n)$输入系统的响应都可表示成其变换 $X(e^{j\omega})$ 和频率响应 $H(e^{j\omega})$ 的乘积形式，方便地在频域中计算。但是，傅里叶变换有两个缺点，其一，在实际中许多有用的信号，如 $u(n)$和 $n*u(n)$……等,它们的离散傅里叶变换不存在。其二，系统对初始条件的暂态响应，或由时变输入引起的系统响应，都无法用离散傅里叶变换方法来计算。

为了克服上述的两个缺点，要把离散傅里叶变换方法进行推广，推广后的方法称为 z 变换。它的单边变换形式可用以获得初始条件和时变输入引起的系统响应；它的双边 z 变换形式提供了另一种领域，可对更多类型的序列和系统进行分析。

5.1.1 z 变换的定义与收敛性

对于任何一个序列 $x(n)$，其双边 z 变换定义为

$$X(z) := Z[x(n)] := \sum_{n=-\infty}^{\infty} x(n) z^{-n} \tag{5.1.1}$$

其中 $z=\rho \cdot e^{j\varphi}$ 是一个复变量，其向径为 ρ，幅角为 ϕ，构成一个复平面，称为 z 平面。而序列 $x[n]$ 的单边 z 变换则定义为：

$$X^{+}(z) := Z^{+}[x(n)] := \sum_{n=0}^{\infty} x(n) z^{-n} \tag{5.1.2}$$

它只对 $x(n)$ 中的正时间序列（以后简称右序列）部分进行运算，而把 $x(n)$ 中的负时间序列（以后简称左序列）部分简单地甩掉了。

例如，对于序列，$x=\{1.5,0,5,\underline{\underline{2}},-2,3,-1\}$即

$$x(n)=1.5\delta(n+2)+0.5\delta(n+1)+2\delta(n)-2\delta(n-1)+3\delta(n-2)-\delta(n-3)$$

它的双边 z 变换是 $X(z)=1.5z^{2}+0.5z+2z^{0}-2z^{-1}+3z^{-2}-z^{-3}$；

可以看出，简单地把 $x(n)$ 乘以 z^{-n}（n=−∞, …, −1, 0,1,…,∞）后相加，就得到了 z 变换。而其单边 z 变换是 $X^{+}(z)=2z^{0}-2z^{-1}+3z^{-2}-z^{-3}$。

如果时间向量 n 只取正值，则序列的 z 变换 $X(z)$中只出现 z 的负次幂；当采样时刻 n 取负值时，$X(z)$中将出现 z 的正次幂。普遍地说，如果 $X(z)$ 只由 z 的正次幂组成，它是左序列。反之，如果 $X(z)$ 只由 z 的负次幂组成，它就是右序列。如果它同时包含了正、负次幂，就是双向序列。如果 $x(n)$ 是右序列，则它的单边 z 变换等同于双边 z 变换。如果 $x(n)$ 是左序列，

则它的单边 z 变换为零。在工程中，人们感兴趣的主要是右序列，即关心信号从当前向未来的变化。即使有时要考虑前一段时间的信号，但由于时间起点选择的任意性，只要序列不是向左无限延伸的，可以把零点左移一个有限整数 n_0，而引入 $\bar{n}=n+n_0>0$，仍然可作为右序列来处理。

根据 z 变换的定义，它是一个无穷级数，无穷级数首先就有一个收敛问题。只有这个级数是收敛的才有意义，才能讨论它的其他性质。下面分类讨论其收敛性。

（1）有限长序列

如果 $x[n]$是有限序列，它只有有限个样本，即 $ns\leqslant n\leqslant nf$，则

$$X(z)=\sum_{n=0}^{\infty}x(n)z^{-n}=\sum_{n=ns}^{nf}x(n)z^{-n}=x(ns)z^{-ns}+x(ns+1)z^{-(ns+1)}+\cdots+x(nf)z^{-nf}$$

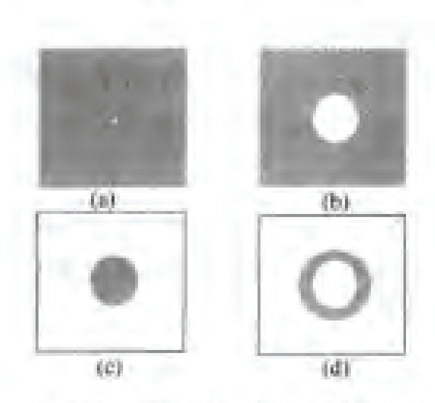


图 5.1.1　各类序列 z 变换的收敛域（灰色部分）

因为它已经不是无穷级数，所以只要序列所有样本的幅度有限，$X(z)$中代入除了原点 z=0 以外的任何 z 值都有确定的值，可见它在除原点以外的整个 z 平面上都是收敛的。把 $z=0$ 代入 $X(z)$会出现无穷大，这个点称为 $X(z)$的极点。此时，$X(z)$的收敛域见图 5.1.1(a)灰色区域。此图形的中心是它的极点，它处于 z 平面的原点，向外可以向无穷远扩展（不限于图中的方块）。

（2）无限长右序列

例 5.1.1：设有一个右序列

$$x(n)=\begin{cases}1.5^n & n=0,1,2,\cdots\\ 0 & n<0\end{cases}\tag{5.1.3}$$

求它的 z 变换

解：因为$1.5^n=\mathrm{e}^{0.4055n}$，此序列随 n 的增长按指数增加到无穷大，因此，它的 DTFT 是发散的，或者说不存在，但却可以有 z 变换。在本题中用双边或单边 z 变换是相同的，它等于

$$X(z):=\sum_{n=0}^{\infty}1.5^n z^{-n}=\sum_{n=0}^{\infty}(1.5z^{-1})^n\tag{5.1.4}$$

这是一个无穷级数，要把这个无穷级数写成紧凑形式，可以利用

$$1+r+r^2+\cdots=\sum_{n=0}^{\infty}r^n=\frac{1}{1-r}\tag{5.1.5}$$

其中 r 是一个复数。根据级数理论，(5.1.5)式仅在$|r|<1$ 时成立。于是（5.1.4）式可以变为

$$X(z)=Z[(1.5)^n]=\frac{1}{1-1.5z^{-1}}=\frac{z}{z-1.5}\tag{5.1.6}$$

这个方程仅在$|1.5z^{-1}|<1$或$1.5<|z|$时成立，也即其收敛区满足$|z|>1.5$。它意味着位于图 5.1.1(b)的 z 平面的阴影线区域内，这个区域从半径为 1.5 的圆周一直扩展到无穷远，也称它在无穷远的邻域收敛。其实对于所有的右序列，只要其样本值是有界的，由于无穷级数中

的通项是 $x(n)z^{-n}$，判断级数是否收敛要看通项的后项域前项之比是否满足

$$|r(n)|=\left|\frac{x(n+1)z^{-(n+1)}}{x(n)z^{n}}\right|=\left|\frac{x(n+1)}{x(n)}z^{-1}\right|<1 \tag{5.1.7}$$

如果 $x(n)$和 $x(n+1)$有界，当判据中的 z 值趋向无穷大时，则$|z^{-1}|$以及$|r(n)|$就趋向无穷小。可见这个级数的收敛性是有保证的。所以 z 变换提供的条件比傅里叶变换宽松得多，特别是在 $z=\infty$附近，几乎所有的信号都可以满足收敛性。可以说，所有实际中见到的信号的 z 变换都存在。

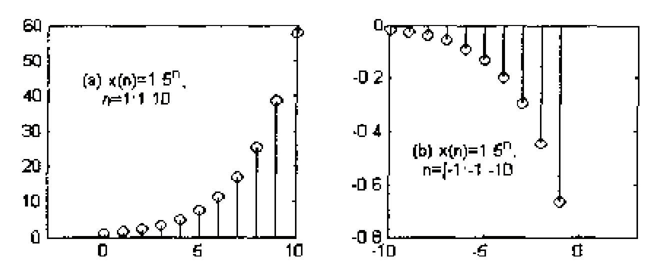


图 5.1.2　例 5.1.1 中的右序列和例 5.1.2 中的左序列

（3）无限长左序列

例 5.1.2：设 $x(n)$为左序列（见图 5.1.2(b)）如（5.1.8）式求其 z 变换。

$$x(n)=\begin{cases}0 & n\geqslant 0\\ -(1.5)^{-n} & n<0\end{cases} \tag{5.1.8}$$

解：因为$(1.5)^{-n}=(0.6667)^{n}$此序列随 n 朝负方向的增长按指数减小。它的单边 z 变换等于零，而双边 z 变换等于下式，其中用$\overline{n}=-n$对 n 做了代换

$$X(z):=\sum_{n=-\infty}^{-1}-1.5^{n}z^{-n}=-\sum_{\overline{n}=1}^{\infty}(1.5^{-1}z)^{\overline{n}} \tag{5.1.9}$$

这个级数的收敛条件应该是$|1.5^{-1}z|<1$，即$|z|<1.5$。这表明，它的收敛域在 z 平面上是半径为 1.5 的中心圆的内部，如图 5.1.1(c)所示。也可以说，它在 z 平面上 $z=0$ 的邻域收敛。左序列之所以不能在 $z=\infty$附近收敛，是因为其通项中出现的是 z^{n}，显然在 $z=\infty$时它趋向于发散，只能在 $z=0$ 的邻域收敛。

由于

$$\sum_{n=1}^{\infty}r^{n}=\sum_{n=0}^{\infty}r^{n}-1=\frac{1}{1-r}-1=\frac{r}{1-r} \tag{5.1.10}$$

$$X(z):=-\frac{1.5^{-1}z}{1-1.5^{-1}z}=\frac{z}{z-1.5} \tag{5.1.11}$$

可以发现这个左序列的 z 变换与例 5.1.1 中右序列的 z 变换（见 5.1.6 式）完全相同。

从这两个例子的比较可以得出结论：双边 z 变换与序列之间，不是一一对应的，它具有双义性。由序列求 z 变换是惟一的；而反过来，由 z 变换求序列（即求 z 反变换）却可以有两个不同的结果，这在工程上会带来困扰。要得到正确的结果，必须要知道它的原始级数的收敛域是在 $z=0$ 的邻域还是在 $z=\infty$的邻域，也就是要知道原始级数中的 z 是正幂还是负幂。更明白地说，必须要知道原始序列是左序列还是右序列。如果排除了对左序列的研究，只研究级数中 z 取负幂的情况，也就是只考虑在 $z=\infty$的邻域收敛的级数，那么双边 z 变换双义性的问题就不存在了。

（4）双向无限序列

正负双向无限序列可以看做一个右序列与一个左序列之和，其 z 变换也是右序列与左序列 z 变换之和。必须两个级数都收敛，它才收敛。因此它的收敛域是图 5.1.1(b)和(c)的相“与”，最后形成一个同心圆环。其外环是左序列的收敛半径，内环是右序列的收敛半径，如图 5.1.1(d)

所示。如果左序列的收敛半径小于右序列的收敛半径，就不存在收敛域。如果排除了对左序列的研究，当然也就排除了对双向无限序列的研究，不需要对它的收敛域再下工夫了。

比较了上述四种情况以后，根据工程中的实际情况，只需考虑图 5.1.1(a)和(b)两种收敛域。前者是对有限序列的，后者是对无限序列的。它们有一个共同的本质的特性，那就是都在 $z=\infty$ 的邻域收敛，并且在这个邻域是解析的。

解析函数的特点之一，是具有各阶导数，因而能展开为幂级数。如果 $X(z)$是由收敛的幂级数组成，把它再展开幂级数是毫无疑义的。而解析函数的初等函数仍然解析。在信号处理中作的加减乘除和超越函数等，都属于初等函数之列。所以只要原始信号在 $z=\infty$的邻域收敛，就可以放心地把系统中所有信号的 z 变换都当做 $z=\infty$的邻域的解析函数对待。也许有的读者对“无穷大的邻域”感到抽象，我们可以把 z 平面看做一个大球，$z=\infty$最后是汇聚球背面的一个点上。在那个点的邻域收敛就是在“无穷大的邻域”收敛，就可以把它在 $z=\infty$的邻域展开为幂级数。

从数学的角度，对于收敛域，还有很多深入的问题可以探讨，例如收敛域的半径如何定，它们的连通区有什么特点等。但从数字信号处理的角度，只要分清所研究的序列是在$|z|=\infty$ 还是在$|z|=0$ 的邻域收敛。收敛域的半径对本课程没有影响，就不深入讨论了。

单边或双边 z 变换的反变换定义为

$$x(n):=Z^{-1}[X(z)]=\frac{1}{2\pi \mathrm{j}}\oint_C X(z)z^{n-1}\mathrm{d}z \tag{5.1.12}$$

其中，积分路径是在复数平面中处于收敛域中的一条围线，积分的方向是反时针的，通常，积分路径选择为 $z=c\mathrm{e}^{\mathrm{j}\omega}$，其中 c 是一个常数，因而是圆心位于 z 平面原点，半径为 c 的一个圆， 如果 c=1，它就是式（3.2.3）中,在 $T=1$ 条件下定义的离散时间傅里叶反变换 IDTFT。在数字信号处理中，不需要用围线积分（5.1.14）式来求 Z 反变换。在第 5.2 节中将专门讨论求 z 反变换的其他方法。

5.1.2　z 变换的重要特性和定理

z 变换的一些重要特性与 DTFT 和 DFT 的特性有很多相似之处，其证明都可以类比或从定义直接导出。这里着重讨论几个重要特性的意义和应用。这部分的证明都限于右序列，所以也免除了对收敛域的说明。以下的部分除非特别注明，所有的 z 变换都在$|z|=\infty$的邻域收敛并在该邻域解析。

（1）线性特性

$$Z[\alpha\, g(n)+\beta\, h(n)]=\alpha\, G(z)+\beta\, H(z) \tag{5.1.13}$$

（2）样本的移位

$$Z[\,g(n-n_0)]=\sum_{n=0}^{\infty}g(n-n_0)z^{-n}=z^{-n_0}\sum_{n=0}^{\infty}g(n-n_0)z^{-(n-n_0)}=z^{-n_0}G(z) \tag{5.1.14}$$

这个关系验证了本书第 2 章中把 z^{-1} 当做移位算子的做法。（5.1.14）式对当时还只是形式上的处理方法给予了严格的理论证明。

在积分变换发展史上，首先出现的是运算微积，由 Heaviside 引入 $D=\dfrac{\mathrm{d}}{\mathrm{d}t}$ 作为微分算子，用代数方法解决了许多微积分的问题，当时并没有严格的证明，后来由 Laplace 提出了拉普

拉斯变换，才在理论上给出了解释。拉普拉斯算子 s 代替了 D。离散序列虽然没有这段历史，但人们从中取得的经验说明，既可以把 z 作为算子，也可以通过 z 变换的理论证明其合理性。

（3）序列卷积

$$\begin{aligned} Z[g(n)*h(n)] &= \sum_{n=-\infty}^{\infty}\left[\sum_{m=-\infty}^{\infty} g(m)h(n-m)\right]z^{-n} = \sum_{m=-\infty}^{\infty} g(m)\left[\sum_{n=-\infty}^{\infty} h(n-m)z^{-n}\right] \\ &= \sum_{m=-\infty}^{\infty} g(m)z^{-m}\left[\sum_{n=-\infty}^{\infty} h(n-m)z^{-(n-m)}\right] = G(z)\cdot H(z) \end{aligned} \tag{5.1.15}$$

序列卷积性质是非常有用的，它把时域中复杂的卷积运算变成了 z 域中 z 的多项式相乘的关系。当一个序列通过离散系统时，输出序列的 z 变换 $Y(z)$，可以用输入序列的 z 变换 $X(z)$ 乘以系统函数 $H(z)$ 求得。$H(z)$就是脉冲响应的 z 变换。当输入是一个脉冲函数 $\delta(n)$时，它的 z 变换为 $X(z)=1$。此时系统的输出 $Y(z)$就是系统函数 $H(z)$，它的时间序列就是 $H(z)$的 z 反变换，后面常常用这个方法来检验 z 反变换的正确性。

（4）初值定理

$$\lim_{z\to\infty} G(z) = \sum_{n=0}^{\infty} g(n)z^{-n}\Bigg|_{z\to\infty} = g(0) \tag{5.1.16}$$

（5）终值定理

若 $x(n)$ 是因果序列，且其 z 变换的极点均在单位圆内部，最多只有一个一阶极点在 $z=1$ 上，则 $x(n)$ 在 n 趋于无穷时的终值可计算如下

$$\lim_{n\to\infty} x(n) = \lim_{z\to 1}(z-1)X(z) \tag{5.1.17}$$

在信号处理中有用的 z 变换特性，主要是上面几个，下面的一些特性可用在信号的解析分析中，但对工程中信号的分析计算，用处不大，以小字列出。

（6）乘以指数序列

$$Z\left[a^n g(n)\right] = \sum_{n=0}^{\infty} a^n g(n)z^{-n} = \sum_{n=0}^{\infty} g(n)\left(\frac{z}{a}\right)^{-n} = G\left(\frac{z}{a}\right) \tag{5.1.18}$$

（7）序列乘以 n (z 变换微分)

$$\frac{d}{\mathrm{d}z}G(z) = \frac{d}{\mathrm{d}z}\sum_{n=0}^{\infty} g(n)z^{-n} = \sum_{n=0}^{\infty} g(n)\frac{d}{\mathrm{d}z}\left(z^{-n}\right) = -z^{-1}\sum_{n=0}^{\infty} ng(n)\left(z^{-n}\right) = -z^{-1}Z\left[n\cdot g(n)\right]$$

因此，$$Z\left[n\cdot g(n)\right] = -z\frac{\mathrm{d}G(z)}{\mathrm{d}z} \tag{5.1.19}$$

（8）时域折叠（此时右序列变为左序列，收敛域由 $z=\infty$邻域变为 $z=0$ 邻域）

$$Z\left[g(-n)\right] = \sum_{n=0}^{\infty} g(-n)z^{-n} = \sum_{\bar{n}=\infty}^{0} g(\bar{n})\left(\frac{1}{z}\right)^{-\bar{n}} = G\left(z^{-1}\right) \tag{5.1.20}$$

（9）复序列共轭

$$Z\left[g^*(n)\right] = \sum_{n=0}^{\infty} g^*(n)z^{-n} = \sum_{n=0}^{\infty}\left[g(n)(z^*)^{-n}\right]^* = \left[\sum_{n=0}^{\infty} g^*(n)(z^*)^{-n}\right]^* = G^*(z^*) \tag{5.1.21}$$

（10）序列乘积

$$Z[g(n)\cdot h(n)]=\sum_{-\infty}^{\infty}g(n)h(n)z^{-n}=\sum_{n=-\infty}^{\infty}\left[\frac{1}{2\pi \mathrm{j}}\oint G(v)v^{n-1}\mathrm{d}v\right]h(n)z^{-n}$$

$$=\frac{1}{2\pi \mathrm{j}}\oint G(v)\sum_{n=-\infty}^{\infty}h(n)z^{-n}v^{n-1}\mathrm{d}v=\frac{1}{2\pi \mathrm{j}}\oint G(v)\left[\sum_{n=-\infty}^{\infty}h(n)\left(\frac{z}{v}\right)^{-n}\right]\frac{\mathrm{d}v}{v}=\frac{1}{2\pi \mathrm{j}}\oint G(v)Y\left(\frac{z}{v}\right)\frac{\mathrm{d}v}{v}$$

由此得到

$$Z[g(n)h(n)]=\frac{1}{2\pi \mathrm{j}}\oint G(v)H(z/v)v^{-1}\mathrm{d}v \tag{5.1.22}$$

5.1.3 z 变换的计算实例

例 5.1.3：设 $x_1(n)=[\underline{2},3,4]$和 $x_2(n)=[\underline{3},4,5,6]$，求它们的 z 变换及两序列的卷积输出 $x_3(n)$。

解：由 z 变换定义，可得 $X_1(z)=2+3z^{-1}+4z^{-2}$ 和 $X_2(z)=3+4z^{-1}+5z^{-2}+6z^{-3}$。为求 $x_1(n)$和 $x_2(n)$的卷积，先求它们的 z 变换的乘积

$X_3(z)=X_1(z)X_2(z)=(2+3z^{-1}+4z^{-2})(3+4z^{-1}+5z^{-2}+6z^{-3})$。

两个 z 负幂的多项式相乘，可以看出其 z^{-1} 最低次为 z^{-0}，系数为 2×3＝6。而最高次为 z^{-5}，系数为 4×6=24。如果要求中间各次,例如 z^{-3} 的系数，可以形象地把第二组系数反过来排列，与第一组系数对齐，如下所示。

```
2      3      4
6      5      4     3
12+    15 +   16+   0= 43
```

把对应项相乘并逐项相加，得到上面的结果 43。动态地看，想像下面那组系数的 3 从最左边对准第一组的 2 的地方开始，算出来的是 6，然后把下面的整组系数右移一位，把对准的各位上下相乘，再把各乘积相加，就得到该项的系数；如此逐次右移，每移一位，就求出一个系数。直移到它的 6 对准第一组的 4 为止，就可依次得出所有的系数。不难看出，多项式相乘时系数的这个运算过程和卷积是非常相似的。所以在 MATLAB 中是用的同一个函数 conv.m。

解这个简单问题的 MATLAB 程序为 x1 = [2,3,4]; x2 = [3,4,5,6]; x3 = conv(x1,x2)。

答案为：x3 ＝ 6 17 34 43 38 24

因此 $X_3(z)=6+17z^{-1}+34z^{-2}+43z^{-3}+38z^{-4}+24z^{-5}$。在这个例题中，从 z 变换的角度考虑，把 $x1$ 和 $x2$ 看做两个 z 变换 $X_1(z)$和 $X_2(z)$的系数向量，两者之间进行多项式的相乘；如果换一种角度，也可以把 $x1$ 和 $x2$ 看做时间序列，两者进行卷积。

例 5.1.4：求长度为 N 的方波函数 $x(n)=R_N(n)$的 z 变换。

解：
$$X(z)=\sum_{n=-\infty}^{\infty}R_N(n)z^{-n}=\sum_{n=0}^{N}z^{-n}=\frac{1-z^{-N}}{1-z^{-1}} \tag{5.1.23}$$

$R_N(n)$是一个因果的有限长序列。当 N 趋向于无穷时，$X(z)=\dfrac{1}{1-z^{-1}}$。这就是阶跃函数的 z 变换。这类常用序列的 z 变换对列于表 5.1.1 中，利用表中的 z 变换对，可以用来查找序列的 z 变换,也可用来根据一些简单的 z 变换找到其对应的序列。这也是求 z 反变换的方法之一。

表 5.1.1　常用的 z 变换对

序　列	z 变 换	收 敛 域
单个脉冲序列 $\delta(n)$	1	全 z 平面
阶跃序列 $\mu(n)$	$\dfrac{1}{1-z^{-1}}$	$\lvert z\rvert > 1$（∞领域）
$-\mu(-n-1)$	$\dfrac{1}{1-z^{-1}}$	$\lvert z\rvert < 1$（0 领域）
$a^{n}\mu(n)$	$\dfrac{1}{1-az^{-1}}$	$\lvert z\rvert > \lvert a\rvert$（∞领域）
$-b^{n}\mu(-n-1)$	$\dfrac{1}{1-bz^{-1}}$	$\lvert z\rvert < \lvert b\rvert$（0 领域）
$[a^{n}\sin\omega_0 n]\mu(n)$	$\dfrac{(a\sin\omega_0)z^{-1}}{1-(2a\cos\omega_0)z^{-1}+a^{2}\cdot z^{-2}}$	$\lvert z\rvert > \lvert a\rvert$（∞领域）
$[a^{n}\cos\omega_0 n]\mu(n)$	$\dfrac{1-a(\cos\omega_0)z^{-1}}{1-(2a\cos\omega_0)z^{-1}+a^{2}\cdot z^{-2}}$	$\lvert z\rvert > \lvert a\rvert$（∞领域）
$na^{n}\mu(n)$	$\dfrac{az^{-1}}{(1-az^{-1})^{2}}$	$\lvert z\rvert > \lvert a\rvert$（∞领域）
$-na^{n}\mu(-n-1)$	$\dfrac{bz^{-1}}{(1-bz^{-1})^{2}}$	$\lvert z\rvert < \lvert b\rvert$（0 领域）

下面举例说明如何利用 z 变换的定义及其性质，以及 z 变换表 5.1.1，来解析地求复杂序列的 z 变换。

例 5.1.5：用 z 变换的性质和 z 变换表，求下行序列的 z 变换。

$$x(n)=(n-2)(0.5)^{(n-2)}\cos[\frac{\pi}{3}(n-3)]u(n-2)$$

解：应用样本移位特性，有 $X(z)=Z[x(n)]=z^{-2}Z[n(0.5)^{n}\cos(\frac{\pi n}{3})u(n)]$。

利用微分特性（5.1.20）式，把乘数 n 从序列运算式中去掉，得到等式

$$X(z)=z^{-2}\left\{-z\frac{d}{\mathrm{d}z}Z[(0.5)^{n}\cos(\frac{\pi}{3}n)u(n)]\right\}$$

由表 5.1.1 查得 $(0.5)^{n}\cos(\frac{\pi}{3}n)u(n)$ 的 z 变换为

$$Z[(0.5)^{n}\cos(\frac{\pi n}{3})u(n)]=\frac{1-(0.5\cos\frac{\pi}{3})z^{-1}}{1-2(0.5\cos\frac{\pi}{3})z^{-1}+0.25z^{-2}}=\frac{1-0.25z^{-1}}{1-0.5z^{-1}+0.25z^{-2}}$$

因此

$$\begin{aligned}X(z)&=-z^{-1}\frac{d}{\mathrm{d}z}\left\{\frac{1-0.25z^{-1}}{1-0.5z^{-1}+0.25z^{-2}}\right\}\\&=-z^{-1}\left\{\frac{-0.25z^{-2}+0.5z^{-3}-0.0625z^{-4}}{1-z^{-1}+0.75z^{-2}-0.25z^{-3}+0.0625z^{-4}}\right\}\\&=\frac{0.25z^{-3}-0.5z^{-4}+0.0625z^{-5}}{1-z^{-1}+0.75z^{-2}-0.25z^{-3}+0.0625z^{-4}}\end{aligned}$$

5.2　z 反变换和差分方程的解

由公式（5.1.13）可知，计算 z 反变换需要在一复围线上求积分，通常这是一个令工程师头痛的过程。实际上并不需要求积分，解决这个问题大体有三种方法。(1) 极点留数法；(2) 部分分式法；(3) 幂级数法（长除法）。MATLAB 为这几种方法提供了简便易用的函数。

5.2.1　用留数定理求 z 反变换

如果 $X(z)z^{n-1}$ 在围线 c 内的极点用 z_k 表示，根据留数定理

$$\frac{1}{2\pi \mathrm{j}}\oint_c X(z)z^{n-1}\mathrm{d}\, z=\sum_k \mathrm{Res}[X(z)z^{n-1},z_k] \tag{5.2.1}$$

式中 $\mathrm{Res}[X(z)z^{n-1},z_k]$ 是被积函数 $X(z)z^{n-1}$ 在极点 $z=z_k$ 处的留数，所以 z 反变换是围线 c 内所有极点的留数之和。各个极点的留数求法如下述。

如果 z_k 是单极点，则

$$\mathrm{Res}[X(z)z^{n-1},z_k]=(z-z_k)\cdot X(z)z^{n-1}\Big|_{z=zk} \tag{5.2.2}$$

如果 z_k 是 N 重极点，则

$$\mathrm{Res}[X(z)z^{n-1},z_k]=\frac{1}{(N-1)!}\frac{d^{N-1}}{\mathrm{d}\, z^{N-1}}\left[(z-z_k)^N\cdot X(z)z^{n-1}\right]\Big|_{z=zk} \tag{5.2.3}$$

例 5.2.1：设 $X(z)=(1-az^{-1})^{-1}$，求其 z 反变换。

解：据公式(5.2.3)，$x(n)=\dfrac{1}{2\pi \mathrm{j}}\oint (1-az^{-1})^{-1}z^{n-1}\mathrm{d}\, z=\dfrac{1}{2\pi \mathrm{j}}\oint \dfrac{z^{n-1}}{1-az^{-1}}\mathrm{d}\, z=\mathrm{Res}\left[\dfrac{z^n}{z-a},a\right]$

因而

$$x(n)=\mathrm{Res}\left[\frac{z^n}{z-a},a\right]=(x-a)\frac{z^n}{z-a}\Big|_{z=a}=a^n \tag{5.2.4}$$

例 5.2.2：设 $X(z)=6z^{-3}$，求其 z 反变换。

解：$x(n)=\dfrac{1}{2!}\dfrac{d^2}{\mathrm{d}\, z^2}\left[(z-0)^3\cdot 6\dfrac{1}{z^3}z^{n-1}\right]\Big|_{z=0}=\dfrac{1}{2}\dfrac{d^2}{\mathrm{d}\, z^2}\left[6z^{n-1}\right]\Big|_{z=0}=3(n-1)(n-2)z^{n-3}\Big|_{z=0}$

按这个式子，当 $n=1,n=2$ 时，x 都是 0。只在 $n=3$ 处，有 $z^{n-3}=0^0=1$，故有 $x(3)=3\times2\times1=6$。本来按照定义，$X(z)=\sum_{n=0}^{3}x(n)z^{-n}=x(0)+x(1)z^{-1}+x(2)z^{-2}+x(3)z^{-3}=6z^{-3}$，用等式左右系数相比较的方法，一眼就可以看出 $x(0)=0$，$x(1)=0$，$x(2)=0$，$x(3)=6$，无须求留数。此例子说明多极点留数计算公式的用法。同时给读者一个概念，当 z 变换中只有分子部分，它们是 z^{-1} 的多项式时，并不是没有极点，而是在原点有多重极点。另外可以看到，留数定理没有给出 $n=0$ 处的 $x(0)$。这是因为 $n=0$ 时，$z=0$ 处并不是极点，所以用留数定理不能解决求 $x(0)$的问题。正确的方法应该是用初值定理（5.1.19）式求 $x(0)$，在本例中是 $x(0)=\lim\limits_{z\to\infty}X(z)=\lim\limits_{z\to\infty}\dfrac{6}{z^3}=0$。

MATLAB 中有求多项式有理分式的极点留数的函数 residuez.m，本书将结合下一节的方法进行介绍。

5.2.2　用部分分式法求 z 反变换

当 z 变换式是一个多项式有理分式时，可以用部分分式分解法，它利用 z 变换表（表 5.1.1 或者其他类似的 z 变换表）进行反变换。在数字信号处理中通常遇到的是这种情况。

若 $X(z)$是 z^{-1} 的有理函数，可用部分分式分解的方法将其变成简单因式项（一阶）的和，由 z 变换表可查出对应于这个因式项的序列，然后再叠加起来。

用部分分式法求 z 反变换过程如下,给定

$$X(z)=\frac{b_0+b_1z^{-1}+\cdots+b_Mz^{-M}}{1+a_1z^{-1}+\cdots+a_Nz^{-N}} \tag{5.2.5}$$

$$X(z)=\underbrace{\frac{\bar{b}_0+\bar{b}_1z^{-1}+\cdots+\bar{b}_{N-1}z^{-N-1}}{1+a_1z^{-1}+\cdots+a_Nz^{-N}}}_{\text{真有理分式}}+\underbrace{\sum_{k=0}^{M-N}C_kz^{-k}}_{\text{若}M\geqslant N\text{，直接多项式}} \tag{5.2.6}$$

式子右边第一项是真有理分式部分，第二项是直接多项式（无穷项）部分。

将真有理分式部分 $X_1(z)$进行部分分式展开，得

$$X_1(z)=\sum_{k=1}^{N}\frac{R_k}{1-p_kz^{-1}} \tag{5.2.7}$$

此处 P_k 是 $X(z)$的第 k 个极点，R_k 是该极点 P_k 处的留数。假设它们都是单极点，则留数 R_k 由下式给出。

$$R_k=\frac{b_0+b_1z^{-1}+\cdots+b_{N-1}\bullet z^{-(N-1)}}{1+a_1z^{-1}+\cdots+a_N\bullet z^{-N}}(1-p_kz^{-1})\bigg|_{z=p_k} \tag{5.2.8}$$

若 p_k 是一个 r 重的极点，则其分解式如下：

$$\sum_{l=1}^{r}\frac{R_{k,l}\bullet z^{-(l-1)}}{(1-p_k\bullet z^{-1})^l}=\frac{R_{k,1}}{1-p_k\bullet z^{-1}}+\frac{R_{k,2}\bullet z^{-1}}{(1-p_k\bullet z^{-1})^2}+\cdots+\frac{R_{k,r}\bullet z^{-(r-1)}}{(1-p_k\bullet z^{-1})^r} \tag{5.2.9}$$

其中留数 $R_{k,f}$ 的计算可使用通用的公式，因太烦琐，此处从略。由于 z 变换多项式已分解为多个分母为 z 的一次项而分子为常数的简单分式，它们的 z 反变换可以从表 5.1.1 中查出。其基本形式为 $Z^{-1}\left[\frac{1}{1-p_kz^{-1}}\right]=p_k^n\mu(n)$。因此，(5.2.7)式的反变换 $x(n)$可求出为：

$$x(n)=\sum_{k=1}^{N}R_kp_k^n\mu(n)+\underbrace{\sum_{k=0}^{M-N}C_k\delta(n-k)}_{M\geqslant N} \tag{5.2.10}$$

可以看出，实现部分分式法并不容易。首先要求出分母多项式的根，以便找到极点。其次，要求出这些极点上的留数。从上例看到，即使是对单极点的反变换也是很烦琐的，所以用计算机取代人工是十分必要的。MATLAB 提供了极点留数计算的两个函数 residue 和 residuez。前者是对拉普拉斯算子 s 的，适用于连续系统；而后者是对 z 变换算子的，适用于离散系统。这里只介绍 residuez，它是信号处理工具箱中的函数。其基本调用格式为[r, p, C]=residuez(b,a)。

其中：b 和 a 为按 z^{-1} 的升幂排列的多项式（5.2.5）的分子和分母的系数向量；

p 为分母的根向量，也就是 $X(z)$ 的极点向量；

r 为对应于根向量中各个根的留数向量；

C 为无穷项多项式系数向量，仅在 $M \geqslant N$ 时存在。

知道了 r，p，C 向量，就可把上述多项式分解为如公式(5.2.11)的形式，所有的常数都成为已知数。

$$\frac{B(z)}{A(z)}=\frac{r(1)}{1-p(1)z^{-1}}+\frac{r(2)}{1-p(2)z^{-1}}+\cdots+\frac{r(N)}{1-p(N)z^{-1}}+C(1)+C(2)z^{-1}+\cdots \qquad (5.2.11)$$

做 z 反变换，得出其时域信号的表达式为(5.2.12)式。

$$y(n)=r(1)p(1)^n\mu(n)+\cdots+r(N)p(N)^n\mu(n)+C(1)\delta(n)+C(2)\delta(n-1)+\cdots \qquad (5.2.12)$$

其中 $\mu(n)$是阶跃函数，在 $n<0$ 时它为 0，在 $n>0$ 时它为 1，保证了 $y(n)$是右序列。

例 5.2.3：计算公式 $X(z)=\dfrac{1}{(1-0.9z^{-1})^2(1+0.7z^{-1})}$ 的反变换。

解：先用函数 poly 求出分母多项式的系数，再用函数 residuez 求 $X(z)$的极点和留数。

```
b=1;a=poly([0.9,0.9,-0.7]);
[r,p,C]=residuez(b,a)
```

得到

```
r =0.2461
   0.5625
   0.1914
p =0.9000
   0.9000
   -0.7000
C = [ ]
```

说明多项式分解为

$$X(z)=\frac{0.2461}{1-0.9z^{-1}}+\frac{0.5625}{(1-0.9z^{-1})^2}+\frac{0.1914}{1+0.7z^{-1}}$$

$$=\frac{0.2461}{1-0.9z^{-1}}+\frac{0.5625}{0.9}z\frac{(0.9z^{-1})}{(1-0.9z^{-1})^2}+\frac{0.1914}{1+0.7z^{-1}}$$

其中第二项化成的形式是为了利用时域移位性（5.1.16）式。根据各项形式，查表 5.1.1 得：

$$x(n)=0.2461(0.9)^n\mu(n)+\frac{0.5625}{0.9}(n+1)(0.9)^{n+1}\mu(n+1)+0.1914(-0.7)^n\mu(n)$$

简化后为 $x(n)=[(0.2461+0.5625)(0.9)^n+0.5625n(0.9)^n+0.1914(-0.7)^n]\mu(n)$ 。

若需要“数值”结果，可以用 MATLAB 计算。比如要求出前八点的值，可以列出：

```
n=0:7; x= (r(1)+r(2))*p(1).^n+r(2)*n. ^p(2). *n+r(3) ^p(3).^n
```

把这条语句和前面的两条语句联在一起执行，得到：

```
x = [1.0000  1.1000  1.6600  1.7540  2.0527  2.1060  2.2459  2.2543 ]
```

如果把 $X(z)$看做一个系统函数，则求 $X(z)$的反变换就得到它的脉冲响应 $h(n)$。求离散系统的脉冲响应用 MATLAB 内部函数 impz.m。其调用方式为 h=impz(b,a,L)。

其中：b,a 为 H(z)的按 z^{-1} 的升幂排列的分子分母系数向量；L 为待求脉冲序列的长度。可以用它来校核计算。在本例中可键入：h=impz(1,poly([0.9,0.9,-0.7]),8)。

得到的结果与上面的 x 相同，只不过输出的向量 h 是沿列向排列的。

例 5.2.4：计算公式 $X(z)=\dfrac{1+0.5z^{-1}}{1-1.2z^{-1}+0.7z^{-2}}$ 的 z 反变换。要求求出的序列是因果的，并且不含复数。

解：从表 5.1.1 中，可以看到当 z 变换的分母由实系数组成而极点是复数时，这些极点将成对共轭出现，这就保证了最后的结果不含虚数。程序 hc524 如下：

```
b=[1,0.5]; a=[1,-1.2,0.7];n=0:7;
[r,p,k]= residuez(b,a)                          % 求极点留数
x=real(r(1)*p(1).^n+ r(2)*p(2).^n),             % 取实部，以避免结果中的微小虚部
```

得到：
```
r = 0.5000 - 0.9432i
    0.5000 + 0.9432i
p = 0.6000 + 0.5831i
    0.6000 - 0.5831i
k = [ ]
```

执行的结果为:

```
x=[1.0000  1.7000  1.3399  0.4180  -0.4364  -0.8163  -0.6740  -0.2375]
```

同样可以用 impz 函数来进行检验，键入：h=impz([1,0.5],[1,-1.2,0.7],8)。

得到的结果与上面的 x 相同。这样看，impz 函数可以当做求反变换的函数看待了。但它是内部函数，因为不知道编写它所依据的理论，所以对于教学的目的而言，只宜把它作为一种检验手段，不提倡直接用于解题。

本题中的极点和留数都是复数，虽然由于它们共轭出现，最后结果应该是实数，但复数运算中的误差会产生一个微小的虚部。所以在第三条语句中加了 real 函数。

在计算过程中，想避免复数运算，可以求出极点的极坐标形式，知道根的幅值和幅角，以便于乘 n 次方，因为这时只要让幅值乘 n 次方，而让幅角乘 n 倍就行了。分别对两个复数根构成的一次项取反变换并相加，就可消去虚部而得到要求的结果。先求出：

alpha=abs(p(1))= 0.8367,

omega=angle(p(1))=0.7711

得知
$$X(z)=\frac{0.5000-0.9432\mathrm{i}}{1-0.8367\mathrm{e}^{\mathrm{j}0.7711}z^{-1}}+\frac{0.5000+0.9432\mathrm{i}}{1-0.8367\mathrm{e}^{-\mathrm{j}0.7711}z^{-1}}$$

查表 5.1.1 中第四行，代入可得
$$x(n)=(0.5-0.9432\mathrm{j})|0.8367|^{n}\mathrm{e}^{-\mathrm{j}0.7711n}\mu(n)+(0.5+0.9432\mathrm{j})|0.8367|^{n}\mathrm{e}^{+\mathrm{j}0.7711n}\mu(n)$$
$$=|0.8367|^{n}[0.5(\mathrm{e}^{-\mathrm{j}0.7711n}+\mathrm{e}^{\mathrm{j}0.7711n})-0.9432\mathrm{j}(\mathrm{e}^{-\mathrm{j}0.7711n}-\mathrm{e}^{\mathrm{j}0.7711n})]\mu(n)$$

再利用指数与正余弦之间的关系，得到
$$x(n)=|0.8367|^{n}[0.5\times2\cos(\mathrm{j}0.7711n)+0.9432\times2\sin(\mathrm{j}0.7711n)]\mu(n)$$

要计算其前八点序列的值，可用下列 MATLAB 语句：

```
n=0:7;
x=alpha.^n.*(real(r(1))*2*cos(omega*n)+imag(r(1))*2*sin(omega*n))
```

可以看出，不利用 MATLAB 的复数运算功能，只能给自己增加麻烦。除非题目要求列出实数的解析式，没有必要在 MATLAB 中避开复数运算。

5.2.3 幂级数法（长除法）求 z 反变换

已经知道，由右序列得到的 z 变换在 $z=\infty$的邻域解析。也就是说，可以把有理分式

$$X(z)=\frac{b_0+b_1z^{-1}+\cdots+b_Mz^{-M}}{1+a_1z^{-1}+\cdots+a_Nz^{-N}}$$

在 $z=\infty$的邻域展开为 z^{-1} 的幂级数。幂级数展开可以用多项式相除的方法实现。其方法是把分子分母的系数都按 z^{-1} 的升幂排列，然后用除法求商。现举例说明如下。

例 5.2.5：求 z 变换 $X(z)=\dfrac{1}{1-1.5z^{-1}}$ 的反变换。

解：采用多项式直接除法，按 z^{-1} 的升幂排列系数，做除法如下：

$$\begin{array}{r l}
 & 1 \;+\; 1.5z^{-1}+(1.5)^2z^{-2}\cdots \\
\cline{2-2}
1-1.5z^{-1}\big) & 1 \;+\; 0 \;+\; \cdots \\
 & 1 \;-1.5z^{-1} \\
\cline{2-2}
 & \quad 1.5z^{-1} \\
 & \quad 1.5z^{-1}-(1.5)^2z^{-2} \\
\cline{2-2}
 & \qquad (1.5)^2z^{-2}\cdots
\end{array}$$

得出其商为幂级数

$$X(z)=1+1.5z^{-1}+(1.5)^2z^{-2}+\cdots \tag{5.2.13}$$

从而求得其反变换为

$$x(n)=\{1,1.5,(1.5)^2,(1.5)^3,\cdots\} \tag{5.2.14}$$

这里要特别注意的是系数向量必须按 z^{-1} 的升幂排列，绝不能按 z 的升幂排列。如果搞反了，按 z 的升幂排列，除式就成为

$$\begin{array}{r l}
 & -0.667z-(0.667)^2z^2-(0.667)^3z^3\cdots \\
\cline{2-2}
-1.5z^{-1}+1\big) & 1 \;+\; 0 \;+\; \cdots \\
 & 1 \;-\; 0.667z \\
\cline{2-2}
 & \quad 0.667z \\
 & \quad 0.667z-(0.667)^2z^2 \\
\cline{2-2}
 & \qquad (0.667)^2z^2\cdots
\end{array}$$

得出的商就成为

$$X(z)=-0.6667z-(0.6667)^2z^2-(0.6667)^3z^3+\cdots\cdots \tag{5.2.15}$$

这实际上成了一个左序列：$x(-1)=0.6667$, $x(-2)=(0.6667)^2$, $x(-3)=(0.6667)^3$,…因为幂级数[见(5.2.15）式]实际上是在 $z=0$ 的邻域展开的，其特点就是幂级数按 z 的升幂排列。

用 MATLAB 实现长除法求 z 反变换：多项式乘法的 MATLAB 函数是 conv.m，除法是乘法的逆运算，它的函数名为 deconv.m。其调用方法为：[q,r] = deconv(b,a)。

其中 b 为分子系数向量，a 为分母系数向量，q 为商的系数向量，r 为余数的系数向量，它们都按 z^{-1} 的升幂排列。如果左端只有一个输出变量，那得出的就是商 q。

这个函数按照 b 向量的长度 M,a 向量的长度 N 进行除法运算。算到余数长度比分母长度小 1 为止。故最后的商 q 的长度为 $M-N+1$,而 r 的长度为 $N-1$。

如果希望商 q 的长度为 Nq，就要通过加零的方法使分子向量 b 的长度增加，使 b 变成一

个新向量 bz，要使得它的长度 Mbz 满足 $Mbz-N+1 \geq Nq$。若取等号，则增加的零的数目应为 $Mbz-M=Nq+N-M-1$。所以长除求反变换的程序的核心语句应为

```
b=?;a=?;Nq(输出向量的长度)=?
M=length(b);N=length(a);
x=deconv([b,zeros(1, Nq+N-M-1)],a)
```

它比部分分式法还要简单得多。而它的理论依据，就是 z 变换在 $z=\infty$处收敛和解析。

例 5.2.6：用长除法重算例 5.2.3，要求求出八点输出。

解：因为 `b=1;a=poly([0.9,0.9,-0.7]); M=1; N=4; Nq=8`。

代入上面的程序（其实就只要一条语句）

```
x=deconv([1,zeros(1,8+4-1-1)],poly([0.9,0.9,-0.7]))
```

运行这条语句得到如下的结果：

```
x = [1.0000  1.1000  1.6600  1.7540  2.0527  2.1061  2.2459  2.2543]
```

它与例 5.2.2 的数值结果完全一样。

例 5.2.7：用长除法重算例 5.2.4，要求求出八点输出。

解：写出 MATLAB 程序 hc526 如下：

```
b=[1,0.5]; a=[1,-1.2,0.7];
N=length(a);M=length(b);Nq=8;
x=deconv([b,zeros(1,Nq+N-M-1)],a)
```

运行结果仍为

```
x = [1.0000  1.7000  1.3399  0.4180  -0.4364  -0.8163  -0.6740  -0.2375]
```

这样看来，求数值结果时，长除法比部分分式法方便得多。部分分式法的好处是能够得到时域信号的解析形式，便于对反变换结果的发展趋势进行定性的分析。读者在应用中要根据问题的不同需求选择适当的方法。

归纳起来，通常遇到的 z 变换只有两种形式，一种是由 z^{-1} 的分子多项式构成：

$$X(z)=\sum_{n=0}^{M} b_a z^{-n}=b_0+b_1 z^{-1}+\cdots+b_M z^{-M}$$

它是有限长右序列的 z 变换，其反变换可以直接写出为

$$x(n)=Z^{-1}[X(z)]=b_0\delta(n)+b_1\delta(n-1)+\cdots\cdots+b_N\delta(n-M)$$

用 MATLAB 的表示式，x 序列即为 $x=[b_0,b_1,\cdots\cdots,b_N]$。可以看出，这与第 2 章中讨论的把 z^{-1} 当做时延算子，直接作用于时域信号具有相同的结果；另一种是由 z^{-1} 的两个多项式之比构成，如下式所示。

$$X(z)=\frac{b_0+b_1 z^{-1}+b_2 z^{-2}+\cdots\cdots+b_m z^{-M}}{1+a_1 z^{-1}+\cdots\cdots+a_n z^{-N}}$$

求它的反变换通常有两种方法：如果只要数值结果，则长除法最为简单，如果要得出时域的表达式，则用部分分式法较好。MATLAB 对这两种情况，都给出了相应的函数。

5.2.4　用 z 变换解差分方程

在第2章中，介绍了线性常系数差分方程的递推解法。本节将研究差分方程的全解，也就是它的全响应。前面说过，z 变换的一个特点是可以求初始条件引起的响应，这时假定输入为零，所以也称为零输入响应：

初始状态为零而加了输入造成的响应称为零状态响应。这两种响应的叠合就是方程的全解。零状态响应可以用卷积来计，前面已经讨论了很多。为了推导初始条件引起的输出，采用单边z变换来分析$n=0$附近的问题。

前节已经给出，序列$x(n)$的单边z变换表示为

$$X^+(z)=Z^+[x(n)]:=Z[x(n)\cdot u(n)]=\sum_{n=0}^{\infty}x(n)z^{-n} \tag{5.2.16}$$

其时移特性如下：

$$Z^+[x(n-k)]:=Z[x(n-k)\cdot u(n)]=\sum_{n=0}^{\infty}x(n-k)z^{-n}=\sum_{m=0}^{\infty}x(m)z^{-(m+k)}$$

$$=\sum_{m=-k}^{-1}x(m)z^{-(m+k)}+\left[\sum_{m=0}^{\infty}x(m)z^{-m}\right]z^{-k}$$

或

$$Z^+[x(n-k)]=x(-1)z^{1-k}+x(-2)z^{2-k}+\cdots+x(-k)+z^{-k}X^+(z) \tag{5.2.17}$$

上面的结果可用于求解具有非零初始条件和非零输入$x(n)$的如下差分方程：

$$\sum_{k=0}^{N}a_k y(n-k)=\sum_{i=0}^{M}b_i x(n-i)\qquad a_0=1,\quad n\geqslant 0 \tag{5.2.18}$$

设其初始条件为　　$y(i),\ i=-1,\dots,\ -N$ 和 $x(i),\ i=-1,\dots,\ -M$。　　(5.2.19)

解的步骤如下：先把方程两端做单边z变换，把方程中的$Z^+[y(n-k)]$和$Z^+[x(n-i)]$都按(5.2.17)展开；然后代入(5.2.18)，整理出$Y^+(z)$的解；最后做Z反变换，得出$y(n)$。下面用例题说明其解法。

例5.2.8：求解

$$y(n)-1.5y(n-1)+0.5y(n-2)=x(n)\quad n\geqslant 0 \tag{5.2.20}$$

其中：$x(n)=(0.25)^n\mu(n)\qquad n\geqslant 0$

初始条件为 $y(-1)=4$ 和 $y(-2)=10$。

解：对差分方程的两边同时进行单边 z 变换，得到

$$Y^+(z)-1.5[y(-1)+z^{-1}Y^+(z)]+0.5[y(-2)+z^{-1}y(-1)+z^{-2}Y^+(z)]=\frac{1}{1-0.25z^{-1}}$$

代入初始条件并整理，得

$$Y^+(z)[1-1.5z^{-1}+0.5z^{-2}]=\frac{1}{1-0.25z^{-1}}+\underbrace{(1.5y(-1)-0.5y(-2)-0.5y(-1)z^{-1})}_{(1-2z^{-1})}$$

或

$$Y^+(z)=\frac{\dfrac{1}{1-0.25z^{-1}}}{1-1.5z^{-1}+0.5z^{-2}}+\frac{(1-2z^{-1})}{[1-1.5z^{-1}+0.5z^{-2}]} \tag{5.2.21}$$

最后得到

$$Y^+(z)=\frac{2-2.25z^{-1}+0.5z^{-2}}{(1-0.5z^{-1})(1-z^{-1})(1-0.25z^{-1})}$$

进行部分分式展开，得到

$$Y^+(z)=\frac{1}{(1-0.5z^{-1})}+\frac{2/3}{1-z^{-1}}+\frac{1/3}{1-0.25z^{-1}}$$

反变换后，得到的解为

$$y(n)=\left[\left(\frac{1}{2}\right)^n+\frac{2}{3}+\frac{1}{3}\left(\frac{1}{4}\right)^n\right]\mu(n) \tag{5.2.22}$$

上面的解是差分方程的全响应。解的形式可以表示为如下几种：

（1）通解和特解

$$y(n)=\underbrace{\left[\left(\frac{1}{2}\right)^n+\frac{2}{3}\right]\mu(n)}_{\text{通解}}+\underbrace{\frac{1}{3}\left(\frac{1}{4}\right)^n\mu(n)}_{\text{特解}} \tag{5.2.23}$$

通解由系统极点确定，特解由输入信号的极点确定。

（2）暂态响应和稳态响应

$$y(n)=\underbrace{\left[\left(\frac{1}{2}\right)^n+\frac{1}{3}\left(\frac{1}{4}\right)^n\right]\mu(n)}_{\text{暂态响应}}+\underbrace{\frac{2}{3}\mu(n)}_{\text{稳态响应}} \tag{5.2.24}$$

暂态响应对应于单位圆内的极点，稳态响应对应于单位圆上的极点。注意，当极点位于单位圆外时，该响应称作无界响应。

（3）零输入响应（或初始条件响应）和零状态响应

式(5.2.21)中，$Y^+(z)$ 由两部分组成：第一部分可以解释为 $Y_{zs}(z)=H(z)X(z)$；第二部分可解释为 $Y_{zI}(z)=H(z)X_{IC}(z)$。

其中 $X_{IC}(z)$可以看做等价的初始条件输入，此输入与初始条件产生同样的输出 Y_{zI} 。此例中，$X_{IC}(n)=\{1,-2\}$。对（5.2.21）式的两部分分别作 z 反变换，全响应可写为

$$y(n)=\underbrace{\left[\frac{1}{3}\left(\frac{1}{4}\right)^n-2\left(\frac{1}{2}\right)^n+\frac{8}{3}\right]u(n)}_{\text{零状态响应}}+\underbrace{\left[3\left(\frac{1}{2}\right)^n-2\right]u(n)}_{\text{零输入响应}} \tag{5.2.25}$$

从本例可以清楚地看出，全响应的三种分解方法分别侧重于系统分析的不同方面，因此分解的结果各不相同，但它们的和都是全响应，是相同的。

下面讲述 MATLAB 实现。在第 2 章中用 filter 函数来解差分方程，需要给出方程的系数和输入。当初始条件给出时，此函数也可用于求全响应。此时，filter 用如下格式调用

```
y=filter(b,a,x,xic)
```

其中，xic 是等效的初始条件输入序列。在例 5.2.8 中，解析地计算 $x_{IC}(n)$。然而在实际中，特别是对于高阶差分方程，解析地计算 $x_{IC}(n)$是很复杂的。MATLAB 提供了 filtic 函数。其调用格式为 xic=filtic(b,a,Y,X)。

其中，b 和 a 是滤波器的分子分母系数数组，Y 和 X 是初始条件数组，分别由 $y(n)$和 $x(n)$的初始条件确定，其形式如下

Y=[y(-1),y(-2),...,y(-N)]

X=[x(-1),x(-2),...,x(-M)]

如果 $n\leqslant-1$ 时 $x(n)=0$，则 filtic 函数中可以缺省 X。对于例 5.2.8，可以这样调用它来确定 $x_{IC}(n)$。键入 Y=[4,10];xic=filtic(b,a,Y)，得到　xic =　　1　　-2

要计算例 5.2.8 的全响应，可以列出程序 hc527 如下：

```
a=[1,-1.5,0.5]; b=1;
n=[0:7]; x=(1/4).^n;
xic=[1, -2];                          % 或用语句Y=[4,10];xic=filtic(b,a,Y)代替
y1=filter(b,a,x,xic)
y2=(1/3)*(1/4).^n+(1/2).^n+(2/3)*ones(1,8) %MATLAB 验算
```

得到 y1 和 y2的结果是相同的，并且与(5.2.23)式得出的响应一致。

例 5.2.9：求解差分方程

$$y(n)=\frac{1}{3}[x(n)+x(n-1)+x(n-2)]+0.95y(n-1)-0.9025y(n-2) \qquad n\geqslant 0$$

其中$x(n)=\cos(\pi n/3)u(n)$，初始条件为$y(-1)=-2; y(-2)=-3; x(-1)=1; x(-2)=1$

要求列出解的函数表达式，然后用数值验证。

解：先解析地求解，然后用 MATLAB 计算。对差分方程作单边 z 变换

$$Y^{+}(z)=\frac{1}{3}[X^{+}(z)+x(-1)+z^{-1}X^{+}(z)+x(-2)+z^{-1}x(-1)+z^{-2}X^{+}(z)]+$$

$$0.95[y(-1)+z^{-1}Y^{+}(z)]-0.9025[y(-2)+z^{-1}y(-1)+z^{-2}Y^{+}(z)]$$

代入初始条件，得到

$$Y^{+}(z)=\frac{(1+z^{-1}+z^{-1})/3}{1-0.95z^{-1}+0.9025z^{-2}}X^{+}(z)+\frac{1.4742+2.1383z^{-1}}{1-0.95z^{-1}+0.9025z^{-2}}$$

显然，$x_{IC}(n)$=[1.4742, 2.1383]，代入 $X^{+}(z)=\dfrac{1.-0.5z^{-1}}{1-z^{-1}+z^{-2}}$ 并化简，得到 $Y^{+}(z)$，它是个有理函数。化简和部分分式展开可以用 MATLAB 来做，程序 hc529 如下：

```
b=[1,1,1]/3; a=[1,-0.95,0.9025];
Y=[-2,-3]; X=[1,1];
xic=filtic(b,a,Y,X)
bxplus=[1,-0.5]; axplus=[1,-1,1];              %X(z)的变换系数
ayplus=conv(a,axplus)                           %Yplus(z)的分母
byplus=conv(b,bxplus)+conv(xic,axplus)          %Yplus(z)的分子
[R,p,C]=residuez(byplus,ayplus)
Mp=abs(p), Ap=angle(p)/pi                       %极坐标形式
```

得到
```
xic =   1.4742   2.1383
     ayplus =   1.0000  -1.9500   2.8525  -1.8525   0.9025
byplus =   1.8075   0.8308  -0.4975   1.9717
R.' =  0.0584 - 3.9468i  0.0584 + 3.9468i  0.8453 + 2.0311i  0.8453 - 2.0311i
p.' =  0.5000 + 0.8660i  0.5000 - 0.8660i  0.4750 + 0.8227i  0.4750 - 0.8227i
C =    [ ]
Mp.' =   1.0000   1.0000   0.9500   0.9500
Ap.' =   0.3333  -0.3333   0.3333  -0.3333
```

为了节省篇幅，此处把列向量 R，p，Mp，Ap 都做了转置，以行向量显示，因此

$$Y^{+}(z)=\frac{1.8076+0.8308z^{-1}-0.4975z^{-2}+1.9717z^{-3}}{1-0.95z^{-1}+2.85251z^{-2}-1.8525z^{-3}+0.9025z^{-4}}$$

$$=\frac{0.0584+\mathrm{j}3.9468}{1-\mathrm{e}^{-\mathrm{j}\pi/3}z^{-1}}+\frac{0.0584-\mathrm{j}3.9468}{1-\mathrm{e}^{\mathrm{j}\pi/3}z^{-1}}+\frac{0.8453+\mathrm{j}2.03118}{1-\mathrm{e}^{\mathrm{j}\pi/3}z^{-1}}+\frac{0.8453-\mathrm{j}2.03118}{1-\mathrm{e}^{-\mathrm{j}\pi/3}z^{-1}}$$

根据表 5.1.1 得

$$y(n)=(0.0584+\mathrm{j}3.9468)\mathrm{e}^{-\mathrm{j}\pi/3}+(0.0584-\mathrm{j}3.9468)\mathrm{e}^{\mathrm{j}\pi/3}$$

$$=(0.8453+\mathrm{j}2.03118)(0.95)^{n}\mathrm{e}^{\mathrm{j}\pi/3}+(0.8453-\mathrm{j}2.03118)(0.95)^{n}\mathrm{e}^{-\mathrm{j}\pi/3}$$

$$=0.1169\cos(\pi n/3)+7.8937\sin(\pi n/3)+(0.95)^{n}[1.6906\cos(\pi n/3)-4.0623\sin(\pi n/3)]$$

$y(n)$的前两项对应稳态响应，后两项是暂态响应。

若只要求出数值结果，可用 filtic 函数，前面已用它求 $x_{IC}(n)$序列。得到的解是数值型的。下面计算 $y(n)$ 的前 8 个点。

```
n=[0:7]; x=cos(pi*n/3);
y=filter(b,a,x,xic)
```

若用它的复数表示式进行验算，键入

```
A=real(2*R(1)); B=imag(2*R(2)); C=real(2*R(3)); D=imag(2*R(4));
y1=A*cos(pi*n/3)+B*sin(pi*n/3)+((0.95).^n).*(C*cos(pi*n/3)+D*sin(pi*n/3))
```

程序运行得到的 y 和 y1 相同。

5.3　z 域中对系统的描述

5.3.1　系统函数（传递函数）

在 z 域中，离散系统的系统函数定义为它的脉冲响应 $h(n)$的 z 变换

$$H(z)=Z[h(n)]=\sum_{n=-\infty}^{\infty}h(n)z^{-n}=\sum_{n=0}^{\infty}h(n)z^{-n} \tag{5.3.1}$$

因为 $h(n)$通常都是右序列，$h(n)=0$ (当 $n<0$)，所以求和的下限可以取为零。

系统的输出序列 $y(n)$等于其输入序列 $x(n)$与系统脉冲响应序列 $h(n)$的卷积，又因 $h(n-k)=0$ (当 $n-k<0$)，有

$$y(n)=\sum_{k=0}^{n}h(n-k)x(k)=\sum_{k=0}^{\infty}h(n-k)x(k) \tag{5.3.2}$$

两边求 z 变换

$$Y(z)=Z[y(n)]=\sum_{n=-0}^{\infty}y(n)z^{-n}=\sum_{n=-0}^{\infty}\left(\sum_{k=0}^{\infty}h(n-k)x(k)\right)z^{-(n-k)}z^{-k}$$

改变求和次序，引入下标变量 $\bar{n}=n-k$, 并利用 $h(n)$ 的因果性，得到

$$\begin{aligned}Y(z)=Z[y(n)]&=\sum_{n=-0}^{\infty}y(n)z^{-n}=\sum_{k=-0}^{\infty}\left(\sum_{n=0}^{\infty}h(n-k)z^{-(n-k)}\right)x(k)z^{-k}\\&=\sum_{k=-0}^{\infty}\left(\sum_{\bar{n}=-k}^{\infty}h(\bar{n})z^{-\bar{n}}\right)x(k)z^{-k}=\left(\sum_{\bar{n}=-k}^{\infty}h(\bar{n})z^{-\bar{n}}\right)\sum_{k=-0}^{\infty}x(k)z^{-k}=H(z)\cdot X(z)\end{aligned} \tag{5.3.3}$$

$Y(z)$，$H(z)$和 $X(z)$之间的关系可以用图 5.3.1 的框图表示。

还可以有另一种等价的定义方法。从第 2 章中知道，LTI 系统的差分方程表示如下：

$$\sum_{k=0}^{N}a_k y(n-k)=\sum_{i=0}^{M}b_i x(n-i)\qquad a_0=1 \tag{5.3.4}$$

对两边取 z 变换。假定系统是因果的，并且初始条件为零（初始松弛的），利用 z 变换的特性，可以得到

$$Y(z)\cdot\sum_{k=0}^{N}a_k z^{-k}=X(z)\cdot\sum_{i=0}^{M}b_i z^{-i} \tag{5.3.5}$$

把系统函数 $H(z)$定义为输出序列的 z 变换 $Y(z)$与输入序列的 z 变换 $X(z)$之比，即

$$H(z)=\frac{Y(z)}{X(z)}=\sum_{i=0}^{M}b_i z^{-i}\Big/\sum_{k=0}^{N}a_k z^{-k}=B(z^{-1})\Big/A(z^{-1}) \tag{5.3.6}$$

$$Y(z)=H(z)X(z)=\frac{b_0+b_1z^{-1}+\cdots+b_Nz^{-N}}{1+a_1z^{-1}+\cdots+a_Nz^{-N}}X(z) \tag{5.3.7}$$

这是系统函数的负幂形式，其中把a_0归一化为 1。在信号处理中，通常都采用这种形式。在一般的线性系统中，系统函数更多地被称为传递函数，不过在信号处理中，多称系统函数。z是定义在一个复数平面上的变量$\rho\,\mathrm{e}^{\mathrm{j}\omega}$，而$H(z)$通常是$z$的多项式或多项式有理分式，所以它也是一个复数。可以用其幅度和相位表示。

如果变量z的取值限于z平面的单位圆圆周上，$\rho=1$，故 $z=\mathrm{e}^{\mathrm{j}\omega}$。则系统函数就成为频率响应：

$$H(\mathrm{e}^{\mathrm{j}\omega})=\sum_{i=0}^{M}b_i z^{-i}\Big/\sum_{k=0}^{N}a_k z^{-k}\Bigg|_{z=\mathrm{e}^{\mathrm{j}\omega}}=\frac{B(\mathrm{e}^{-\mathrm{j}\omega})}{A(\mathrm{e}^{-\mathrm{j}\omega})} \tag{5.3.8}$$

所以，频率响应是 z 变换表示的系统函数的特殊情况。或者说傅里叶变换是 z 变换的特例。为了方便，常常统称为系统函数，只是用括号中的自变量取 z 或取$\mathrm{e}^{\mathrm{j}\omega}$（也可以直接用$\omega$）加以区分。

这样，有四种表示离散系统特性的方法：差分方程；脉冲响应；频率响应；系统函数；它们之间的转换关系列成表 5.3.1。原始系统的表示方法在最左列，变换后系统的表示方法在第一行，变换方法列于表中。例如已知的是系统的脉冲响应，就看表的第二行。若要求频率响应,可用 DTFT[$h(n)$]；若要求系统函数，可用 Z[$h(n)$]；若要求差分方程，如果已知脉冲响应是有限长序列(FIR)，差分方程的系数$b(n)$就与脉冲响应$h(n)$一一对应；但如果已知脉冲响应是无限长序列(IIR)，求它的差分方程系数就是一个相当难的命题。到现在还没有完全解决。可以利用这个表，归纳已学过的内容，并可在方框中补上可以采用的 MATLAB 函数。

表 5.3.1　表示离散系统特性的方法及其相互关系

	差分方程	脉冲响应 $h(n)$	频率响应 $H(\mathrm{e}^{\mathrm{j}\omega})$	z 系统函数 $H(z)$
差分方程		递推， impz(b,a), filter(b,a,impseq)	DTFT 后，解 Y/X Freqz(b,a,w)	z 变换，解 Y/X $H(z)=B(z\text{-}1)/A(z^{-1})$ 用移位算子 z^{-1}
脉冲响应	FIR 情况: $b(n)=h(n)$ IIR 情况:困难		$H(\mathrm{e}^{\mathrm{j}\omega})$ =DTFT[$h(n)$]	z 变换 $H(z)$=Z[$h(n)$]
频率响应 $H(\mathrm{e}^{\mathrm{j}\omega})$	分子、分母系数对应于差分方程右、左系数	IDTFT[$H(\mathrm{e}^{\mathrm{j}\omega})$]		$\mathrm{e}^{\mathrm{j}\omega}=z$
z 系统函数 $H(z)$	$H(z)$用 z^{-1}表示，展开时用时延代替 z^{-1}	z 反变换	$z=\mathrm{e}^{\mathrm{j}\omega}$	

用了z变换表示的系统函数$H(z)$,可以把 LTI 系统的输入输出关系用图 5.3.1(a)表示，此时$Y(z)=H(z)X(z)$。其中$Y(z)$和$X(z)$是输入输出序列的z变换。在实际处理信号的许多框图中，人们往往不用$Y(z)$和$X(z)$而直接采用了序列$y(n)$和$x(n)$自身，画成图 5.3.1(b)，这种直接对序

列信号进行处理的框图，来源于（2.4.18）式，它对于设计计算机流程特别直观。但应该注意的是图 5.3.1 中，当 $H(z)$含有分母部分时是无意义的，因为这是无法运算的。

（a）在 z 域的输入输出关系

（b）时域的输入输出关系（H 中不含除法）

图 5.3.1　系统函数与输入输出的关系

下面举例说明在信号通过系统时，如何用系统函数 $H(z)$与输入信号 z 变换 $X(z)$相乘而求输出 $Y(z)$的问题。

例 5.3.1：设系统函数为 $H(z)=\dfrac{-3z^{-1}}{2-5z^{-1}+2z^{-2}}$，输入信号 $x=[2,4,3,5]$，用 z 变换计算输出 $y(n)$。

解：先求输入信号的 z 变换 $X(z)$，将它乘以系统函数 $H(z)$ 得到输出信号的 z 变换 $Y(z)=H(z)\cdot X(z)$，最后对 $Y(z)$ 求 z 反变换得到 $y(n)=Z^{-1}[Y(z)]$。由给定条件知：

$$X(z)=2+4z^{-1}+3z^{-2}+5z^{-3}，故 Y(z)=H(z)X(z)=\frac{-3z^{-1}}{2-5z^{-1}+2z^{-2}}X(z)=\frac{B(z)}{A(z)}。$$

即系统的 b=-3,a=[2, -5,2]，x=[2,4,3,5]。注意，多项式乘法用 conv 函数， B=conv(b,x), A=a, 由[r, p, k]= residuez (B, A) 可由 B, A 求出 r, p, k，进而求 Z 反变换，得出

$$y(n)=r(1)p(1)^n\mu(n)+r(2)p(2)^n\mu(n)+k(1)\delta(n)+k(2)\delta(n-1)$$

其中 $u(n-n_0)$ 和 $\delta(n-n_0)$分别为在 n_0 处的单位阶跃函数及单位脉冲函数，可由本书提供的 stepseq 和 impseq 函数求得。由此可编成 MATLAB 程序 hc531 如下：

```
x=[2,4,3,5];                          % 输入序列及初始时间
nfx=length(x)-1;                      % 计算序列终止时间
b=-3;a=[2,-2.2,0.5];                  % 系统函数的分子分母系数,
B=conv(-3,x);A=a;                     % 输入与分子 z 变换的多项式乘积,分母不变
[r,p,k]=residuez(B,A)                 % 求留数 r,极点 p 及直接项 k
nf=input('终点时间 nf= ');            % 要求键入终点时间
n = 0:nf;                             % 生成总时间数组
% 求无限序列 yi 和直接序列 yd
yi=(r(1)*p(1).^n+r(2)*p(2).^n).*stepseq(0,0,nf);
yd =k(1)*impseq(0,0,nf)+k(2)*impseq(1,0,nf);
y=yi+yd;                              % 合成输出
绘图语句从略。
```

程序运行结果为

```
nsy =-1
r =  -57.7581
     204.7581
p =   0.7791
      0.3209
k =  -150   -30
```

由此写出 $X(z)$的部分分式如下：

$$Y(z)=z\left(\frac{r(1)}{1-p(1)z^{-1}}+\frac{r(2)}{1-p(2)z^{-1}}+k(1)+k(2)z^{-1}\right)$$

所以，得到 $x(n)$ 表达式：

$$y(n)=-57.7581\cdot 0.7791^{n-1}u(n-1)+204.7581\cdot 0.3209^{n-1}u(n-1)-150\cdot\delta(-1)\ -30\cdot\delta(0)$$

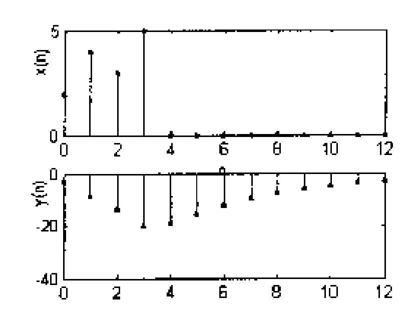


图 5.3.2　例 5.3.1 的输入和输出

$x(n)$和 $y(n)$的图形见图 5.3.2。如果不需要求出表达式，那本题就可以直接用工具箱函数 filter 来解，键入

```
x=[2,4,3,5,zeros(1,10)];b=-3;a=[2,-2.2,0.5];
y=filter(b,a,x); stem([0:ny], stem([0:ny],y)
```

所得 y 的波形与图 5.3.2 相同，而输出 y 长度则取决于输入 x 的长度。为了得到足够的输出点数，在给定的四点输入序列 x 的后面加了 10 个零。

5.3.2　正幂系统函数和零极增益形式

把分子分母同乘以 z^N（此处设 $N \geq M$，若 $N \leq M$，则同乘以 z^M），可以得到系统函数的正幂形式：

$$H(z)=\frac{b_0 z^{N-M}(z^M+\frac{b_1}{b_0}z^{M-1}+\cdots+\frac{b_M}{b_0})}{z^N+a_1 z^{N-1}+\cdots+a_N}=\frac{\overline{B}(z)}{\overline{A}(z)} \tag{5.3.9}$$

一般地，其分子分母系数向量都会发生变化 $B \neq \overline{B}, A \neq \overline{A}$，主要是前面或后面会增加几个零。把正幂形式的系统函数分子分母各自分解因式后，得到

$$H(z)=b_0 z^{N-M}\frac{\prod_{l=1}^{M}(z-z_l)}{\prod_{k=1}^{N}(z-p_k)} \tag{5.3.10}$$

由此式，$H(z_l)=0$，故 z_l 定义为系统的零点；$H(p_k)=\infty$，p_k 就定义为系统的极点；b_0 则相当于增益。因此，这种形式的系统函数称为零极增益形式。可见，一个 LTI 系统可在 z 域中用零极点图的形式来描述。这个事实在设计简单的滤波器时很重要，只要正确地配置零极点就可达到一定的设计要求。用负幂传递函数也可以获得相应的零极点表达式，其形式为(5.3.11)式。

$$H(z)=b_0\frac{\prod_{l=1}^{M}(1-z_l\cdot z^{-1})}{\prod_{k=1}^{N}(1-p_k\cdot z^{-1})} \tag{5.3.11}$$

注意自变量 z 和 z^{-1} 的不同。而且零点 z_l 和极点 p_k 都出现在 z^{-1} 的乘数上。

分别对正幂有理函数 $H(z)$的分子分母多项式使用 MATLAB 的 roots 函数，就可求得其零极点。roots 的逆向函数是 poly，用它可通过多项式的根求得其系数。用 MATLAB 的 zplane(b,a) 函数，也可由给定的分子行向量 b 和分母行向量 a 绘制成系统的零极点图。作图时，通常用点型符号“o”表示零点，“x”表示极点。图中还会自动给出用作参考的单位圆。类似的，zplane(z,p)通过列向量 z 和列向量 p 来绘制零极点图。在使用这个函数时，要注意输入变元用行向量和列向量会得出完全不同的结果，行向量是指多项式系数向量，列向量是指多项式零极点向量。

例 5.3.2：设系统的差分方程为：

$$y(n)+0.3y(n-1)=2x(n)+4x(n-2)-2x(n-4)$$

求此系统的系统函数，分别求出它的负幂形式、正幂形式和零极增益形式。

解：对两边做 z 变换，得到 $Y(z)+0.3z^{-1}Y(z)=2X(z)+4z^{-2}X(z)-2z^{-4}X(z)$。

所以其系统函数的负幂形式为 $H(z)=\dfrac{Y(z)}{X(z)}=\dfrac{2+4z^{-2}-2z^{-4}}{1+0.3z^{-1}}=\dfrac{B(z^{-1})}{A(z^{-1})}$。

它的系数向量为 $a=[1,0.3];b=[2,0,4,0,-2]$。若化成系统函数的正幂形式，分子分母同乘以 z^4，得到

$$H(z)=\frac{z^4}{z^4}\cdot\frac{2+4z^{-2}-2z^{-4}}{1+0.3z^{-1}}=\frac{\overline{B}(z)}{\overline{A}(z)}=\frac{2z^4+4z^2-2}{z^4+0.3z^3}$$

因此，它的系数向量为 $\overline{a}=[1,0.3,0,0,0];\overline{b}=[2,0,4,0,-2]$。

可以看到两种形式的系数向量是不同的。因为正幂形式的系数向量从 z 的最高幂一直排到零次幂（常数项），低次项有一个空缺时要在系数向量中放一个 0。而负幂形式的系数向量从 z 的零次幂（常数项）一直排到最低幂（负的最高幂），最低幂以后就不管了，因此尾部不会出现 0。在应用 MATLAB 函数时，这一点需要特别注意，因为 MATLAB 中，对连续系统开发的函数，一般用正幂形式的系数向量 a 和 b；对离散系统开发的函数，一般用负幂形式的系数向量 a 和 b。信号处理工具箱的绝大部分函数，若用到系数向量 a 和 b,通常都是用负幂形式。但有些公共的函数，比如求根的函数 roots 与 m，它首先是对连续系统开发出来的，以后又推广应用于离散系统，所以它用的就是正幂形式的系数向量。

现在回到对例 5.3.2 求解。先用 MATLAB 对正幂系统函数的分子分母分别求根如下。

```
z=roots([2,0,4,0,-2]),p=roots([1,0.3,0,0,0])
```

得到
```
z = -0.0000 + 1.5538i
    -0.0000 - 1.5538i
    0.6436
    -0.6436
p = [0; 0; 0; -0.3]
```

故其系统函数的零极增益形式为 $H(z)=\dfrac{2(z-1.5538j)(z+1.5538j)(z-0.6436)(z+0.6436)}{z^3(z+0.3)}$。

此系统函数分子分母中 z 的最高次均为 4，有四个零点和四个极点，增益项为 2。键入以下语句，可以看到系统零极点在 z 平面上的分布，如图 5.3.3 所示。

```
a=[1,0.3,0,0,0];b=[2,0,4,0,-2];
zplane(b,a)     % 画零极点分布图
```

注意，系数 a 和 b 作为系数时必须用行向量输入。如果输入的是零极点向量 z 和 p，那么它们必须取列向量形式。zplane 函数就是靠行向量、列向量的不同来理解用户意图的。

用负幂形式也能表示零极点形式。本例中系统函数可以写成

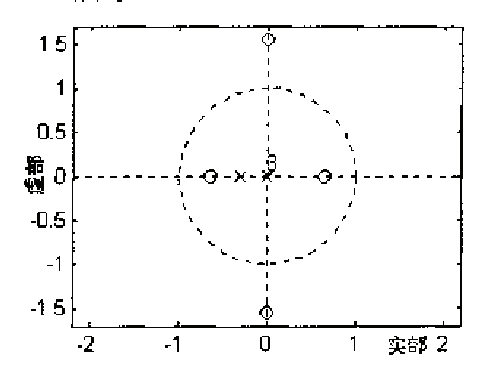


图 5.3.3　系统函数零极点在 z 半面上的分布

$$H(z^{-1})=0.5\cdot\frac{(1-1.5538\mathrm{j}z^{-1})(1+1.5538\mathrm{j}z^{-1})(1-0.6436z^{-1})(1+0.6436z^{-1})}{1+0.3z^{-1}}$$

式中系统的零极点可以看得很清楚，只是在原点的三重极点没有明确的显示。因为分子上的 z^{-1} 的最高次幂是 z^{-4}，而分母上的 z^{-1} 的最高次幂是 z^{-1}，当 z 趋于 0 时，z^{-1} 趋于无穷，分子分母相约以后，分子上还多一个 z^{-3}，它就代表了在 z=0 处的三重极点。从 z 的正幂换成负幂，零极点的位置虽然没有变化，但增益却成了倒数。为了取得统一的标准，零极增益的正规形式是按 z 的正幂定义的。

前面已经指出 MATLAB 中与零极点有关的函数，凡是由连续系统扩展而来的，大多数都用 z 的正幂定义。比如由传递函数系数变换为零极增益的函数[z,p,k]=tf2zp(b,a)和它的逆函数[b,a]=zp2tf(z,p,k)，又如求离散系统脉冲响应和阶跃响应的函数[h,t]=dimpulse(b,a)和[h,t]=dstep(b,a)等，其中的 b 和 a 都按 z 的正幂多项式定义。

有些函数是由信号处理专家开发的，如 impz(b,a)就与 dimpulse 函数等价，用的是负幂系数的 b,a。又如 residuez.m 函数虽然是从连续系统的 residue.m 函数扩展而得的，它也是用的负幂系数。避免出错的一个办法是把离散系统的分子分母系数的长度取得一致，把它们的同次幂系数排在同样位置，如上例中的 a 和 b 的取法。缺项的系数赋值为零，如果分母比分子 z 的阶次高，b 的前几项赋值 0。这样不管正幂和负幂，所得结果都是一样的。总之，MATLAB 工具箱中的 a 和 b 可能有不同含义。它不像书上那样，可以在上面加短划线来区分，调用函数时，应尽量先用 help 命令确实弄清其输入变元的意义和要求。

现在回到由 zplane 函数生成的图 5.3.3。用 “x” 表示此系统的四个极点位置，在原点有一个三重极点；又用 “o” 表示此系统的四个零点位置，其中两个在实轴上，另两个是在虚轴上的一对共轭虚数零点。本例中，分子分母多项式的最高正幂次数相同，所以极点和零点的数目相等。若分母的最高正幂次数高于分子，零极点图上看到的极点数目比零点多。实际上这意味着在 $z=\infty$ 处还有零点，$H(\infty)=0$。把这些无穷远处的零点也算上，系统的零极点数目总是相同的。因为对右序列，$H(z)$在 $z=\infty$ 处收敛，所以在该处只会有零点，不可能有极点。

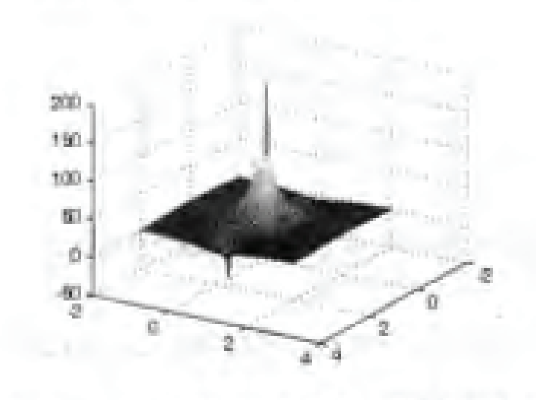


图 5.3.4　$H(z)$的对数幅特性在 z 平面上的分布

利用 MATLAB 的三维图形功能，可以画出系统函数 $H(z)$的对数幅特性在 z 平面上的分布，如图 5.3.4 所示。从图上可以大体看出零点和极点的位置，它是用下列 MATLAB 语句得到的。

```
a=[1,0.3,0,0,0];b=[1,0,2,0, -1];
[X,Y] = meshgrid([-2:0.05:2]+0.001,[-2:0.05:2]+0.001);   % 设 z 平面自变量矩阵
H=polyval(b,X+j*Y)./polyval(a,X+j*Y);                     % 求全 z 平面上的 H(z)
mesh(Y,X,20*log10(abs(H)))
```

程序第二行语句是为了设置平面自变量矩阵 X 和 Y，每个方向各 81 个点，自变量总数为 81×81=6561 个点，在 meshgrid 函数中加的 0.001 是为了避开真正的极点，如果真出现 $H(z)=\infty$，该图就画不出来了。第三行语句是为了求 $H(z)$，在 MATLAB 的信号处理工具箱中的求频率特性的函数 freqz 只能求单位圆上的 $H(z)$，不能求全 z 平面的 $H(z)$。所以只能从 MATLAB 基本的多项式求值函数 polyval.m 出发，若要求出多项式 $Y=p(1)x^N+p(2)x^{N-1}+\cdots+p(N)x+p(N+1)$ 的值，调用它的格式为 Y=polyval(p,x)。

其中：p 为长度为 $N+1$ 的系数向量；x 为自变量的值。x 也可以是数组或矩阵，求出的就是一系列自变量下的多项式值。把 $X+j*Y$ 作为自变量代入，分别求出分子分母多项式的值并相除，就得到 z 平面上很多点的 $H(z)$。注意，这里用的都是元素群运算，polyval 求出的是 6561 个分子和分母复数值，再经点除得出 6561 个 $H(z)$。第四行中的 mesh 函数是用来画三维图形的，它以 X 和 Y 为平面自变量，在 z 方向画 $H(z)$的模（幅度）。之所以对 $H(z)$取对数幅度，是为了突出地显示零点，因为零的对数趋向负无穷大。

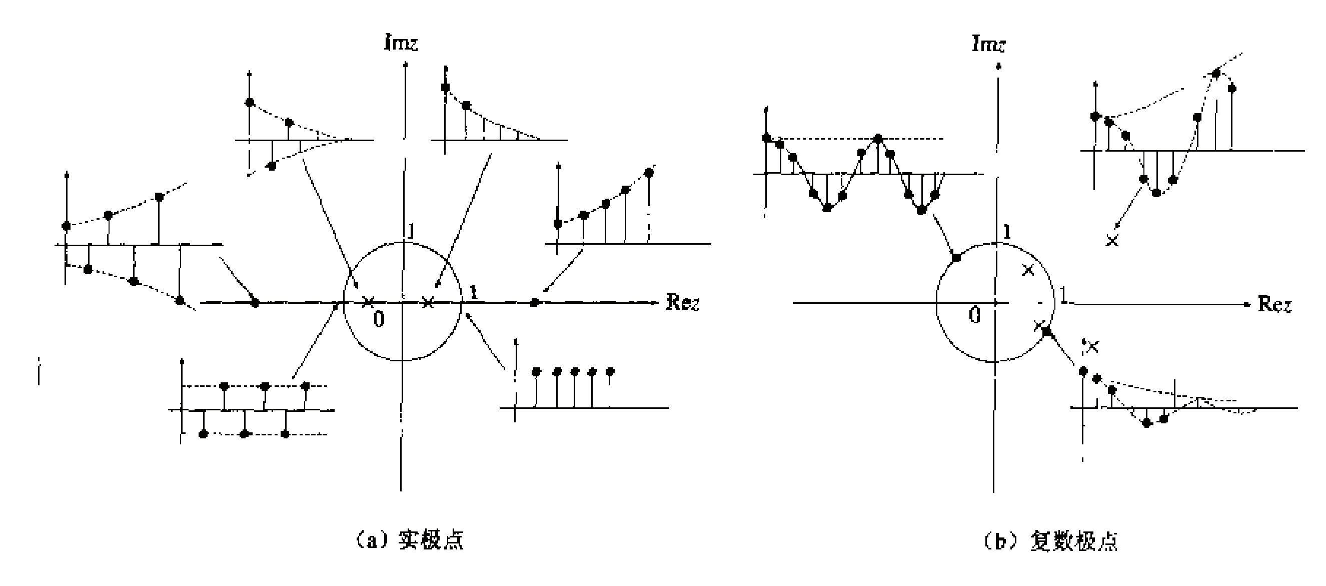


图 5.3.5　不同的极点位置对应的系统响应

系统的极点位置对系统的响应有着特别明显的影响。图 5.3.5 画出了不同的极点位置对应的系统响应，可以看出如下四项共同规律。

（1）在单位圆内的单极点和重极点：当 $n\to\infty$时，脉冲响应趋向于零。

（2）在单位圆外的单极点和重极点：当 $n\to\infty$时，脉冲响应趋向于无穷大。

（3）在单位圆上的单极点：当 $n\to\infty$时，脉冲响应趋向于常数和等幅振荡。

（4）在单位圆上的重极点：当 $n\to\infty$时，脉冲响应趋向于无穷大。

所以只有当系统的极点位于单位圆内的情况下，系统的响应才会在 $n\to\infty$时趋向于零。

5.3.3　零极点分布与系统的因果性和稳定性

（1）因果性

第 2 章中曾用脉冲响应的特性来判断系统的因果性和稳定性，现在再从变换域中讨论这个问题。在时域中，因果系统的充分必要条件是当 $n<0$ 时，$h(n)=0$。这就是说，系统的脉冲响应必须是右序列。前面指出，右序列在变换域的特征是它的 z 变换在无穷远处收敛。因此因果系统的极点不可能在无穷远处。从前面的例子也看到，系统的零点有可能在无穷远处，但极点则不能在无穷远处，只能在 z 平面上一个有界的以原点为中心的圆内。

所以在变换域判断一个因果系统，可以根据它的 z 变换在无穷远处没有极点作为准则。也可用系统在有界范围内的极点的数目不小于零点的数目为准则。如果系统函数用的是正幂形式，则因果性的要求是 $N\geqslant M$，即分母上 z 的最高次数大于分子上 z 的最高次数。

如果系统函数用的是负幂形式，分子分母相约后，分母的起始项必须是常数项（零次幂），分子的起始项为 z^{-k}, $k=0$ 或正整数。即分子上可以有剩余的 z^{-1}，这就能保证当 $z\to\infty$时，$H(z)$

→常数或 0，也就保证了它的因果性。

（2）稳定性

第 2.5.5 节在时域讨论了系统的稳定性判据，系统稳定的充分必要条件是它的脉冲响应绝对可加：

$$\sum_{n=0}^{\infty}|h(n)| = B_h < \infty \tag{5.3.12}$$

从变换域的角度看，离散 LTI 集总参数系统的传递函数 $H(z)$总能用两个有理多项式的分式来表示，因此，其所有极点都位于单位圆之内，是保证它稳定的充分必要条件。这可以证明如下。

因为脉冲响应是 $H(z)$的反变换，把 $H(z)$展开为部分分式，成为各单极点分量的分式和。如果有一个极点在单位圆外，它所对应的反变换时间分量就会随 n 增加而趋于无穷大。如果 $h(n)$中有一项不满足绝对可加条件，系统就不可能稳定。所以必须使 $H(z)$的所有极点都位于单位圆之内。

可以证明，如果系统是稳定的，那么 $H(z)$在 z 平面的单位圆上取值有界，因而其频率特性有定义。事实上，因为$\left|e^{-jn\omega}\right|=1$，所以对所有的 n 和 ω有

$$\left|H(e^{j\omega})\right| = \left|\sum_{i=0}^{M} h(n)\,e^{-jn\omega}\right| \leqslant \sum_{i=0}^{M}|h(n)|\cdot\left|e^{-jn\omega}\right| = \sum_{i=0}^{M}|h(n)| < \infty \tag{5.3.13}$$

在此情况下，由于计算 z 反变换时的积分路径可选 $c=1$ 的单位圆，所以对稳定系统，z 反变换和傅里叶反变换的结果相同。

$$Z^{-1}[H(z)] = F^{-1}[H(e^{j\omega})] \tag{5.3.14}$$

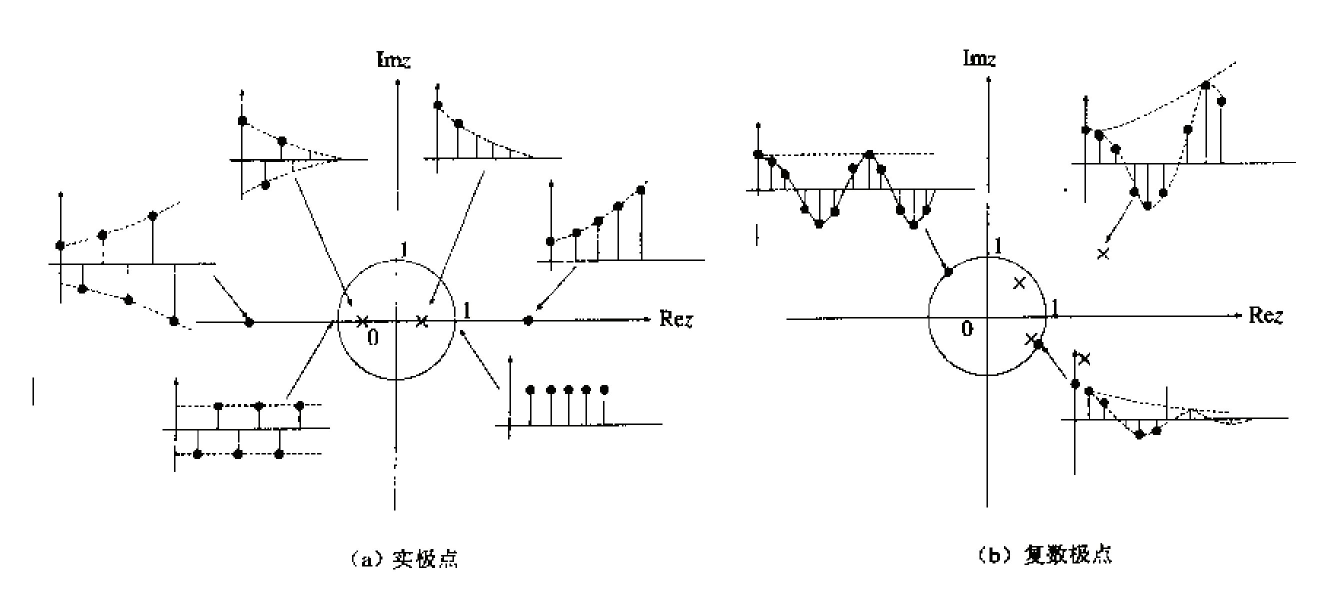


图 5.3.5　不同的极点位置对应的系统响应

系统的极点位置对系统的响应有着特别明显的影响。图 5.3.4 画出了不同的极点位置对应的系统响应，可以看出如下四项共同规律。

（1）在单位圆内的单极点和重极点：当 $n\to\infty$时，脉冲响应趋向于零。

（2）在单位圆外的单极点和重极点：当 $n\to\infty$时，脉冲响应趋向于无穷大。

（3）在单位圆上的单极点：当 $n\to\infty$时，脉冲响应趋向于常数和等幅振荡。

（4）在单位圆上的重极点：当 $n\to\infty$时，脉冲响应趋向于无穷大。

所以只有当系统的极点位于单位圆内的情况下，系统的响应才会在 $n\to\infty$时趋向于零。

5.4　z 平面上的谱分析

第 3.4 节中已经讨论过离散 LTI 系统的频率特性问题。在学了 z 变换以后，可知傅里叶变换是 z 变换的特例，z 变换在整个 z 平面上取值，而傅里叶变换仅在 z 平面上的单位圆上取值。因此，现在可以站在更高、更普遍的角度来再认识频率特性。

5.4.1　零极点与频率特性的关系

如果单位圆的圆周处在系统函数 $H(z)$的收敛域内，也就是说，系统的极点都处在单位圆的内部（因而系统是稳定的），这时就可以计算这个单位圆圆周上的 $H(z)$。在单位圆周上，$z=1\cdot e^{j\omega}$，所以其频率特性可表示为（5.3.5）式。用零极增益方法，可写成

$$H(e^{j\omega})=b_0 e^{j(N-M)\omega}\frac{\prod_{l=1}^{M}(e^{j\omega}-z_l)}{\prod_{k=1}^{N}(e^{j\omega}-p_k)} \tag{5.4.1}$$

因子$(e^{j\omega}-z_l)$可解释为 z 复平面中由零点 z_l 指向单位圆周上 $z=e^{j\omega}$ 处的向量，同样因子$(e^{j\omega}-p_k)$可表示为 z 复平面中由极点 p_k 指向单位圆周上 $z=e^{j\omega}$ 处的向量，如图 5.4.1 示。因此幅值频率响应函数如下式：

$$\left|H(e^{j\omega})\right|=b_0\frac{\left|(e^{j\omega}-z_1)\right|\left|(e^{j\omega}-z_M)\right|}{\left|(e^{j\omega}-p_1)\right|\left|(e^{j\omega}-p_N)\right|} \tag{5.4.2}$$

上式可看成所有零点到单位圆周上给定点的诸矢量长度的积，除以所有极点到该点诸矢量长度的积，再乘以 b_0。类似地，相频响应函数可以看成是三个部分的线性和，它们分别是：常数项、线性相位项与非线性相位项。后者指的是上述诸矢量的相角代数和，如（5.4.3）式右端所示。

$$\angle H(e^{j\omega})=\underbrace{0或\pi}_{常数项}+\underbrace{(N-M)\omega}_{线性项}+\underbrace{\sum_{l=1}^{M}\angle(e^{j\omega}-z_l)-\sum_{k=1}^{N}\angle(e^{j\omega}-p_k)}_{非线性项} \tag{5.4.3}$$

例 5.4.1：举一个数字实例来说明。假如系统有一对位于 $0.5\pm0.7j$ 的共轭极点，另有位于 0.3 和 1.5 的两个实数零点，增益为 2。则其系统函数为

$$H(e^{j\omega})=\frac{2(e^{j\omega}-0.3)\cdots(e^{j\omega}-1.5)}{(e^{j\omega}-0.5-0.7j)\cdots(e^{j\omega}-0.5+0.7j)}$$

用 zplane([0.3;1.5],[0.5+0.7j,0.5−0.7j])语句，它的零极点如图 5.4.1 所示。图上用短虚线画出的圆是单位圆，长虚线的圆是收敛域的内边界，它紧靠两个极点又不包含极点，所以收敛域的内边界是开边界。收敛域的外边界则是无穷远，所以这个系统的脉冲响应是右序列，是因果的。而且因为它的极点都在单位圆内，所以它是稳定的。

从图 5.4.1 上可以分析系统频率响应的几何意义。设要求出系统在数字频率 ω 处频率响应，在单位圆上取相角为 ω 的点，把两个极点和两个零点都向该点连上向量。系统的幅特性应该等于两个零点到它的向量（虚线）的乘积除以两个极点到它的向量（实线）。系统的相特性也等于两个零点向量的相角之和减两个极点向量的相角之和。要特别注意，如果极点靠单位圆很近时，频率特性在靠近该极点附近会出现大的谐振峰，这是由于极点到它的距离很近，分母迅速减小的缘故。

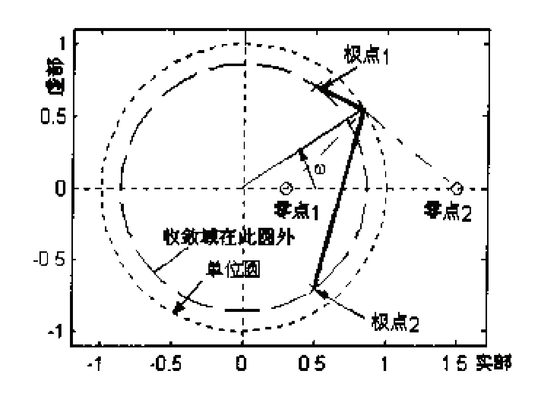


图 5.4.1　频率特性与零极点的几何关系

从零点的分布还可以导出最小相位系统的概念。若零点在单位圆内，当 ω 沿反时针方向由 0 增加至 π 时，由零点产生的向量 β 的相角 ψ 也由零增加 π；若零点在单位圆外左方，则向量 β 的相角 ψ 先由零增加到某个小于 $\pi/2$ 的正数，然后又减小到零；若零点在单位圆外右方，则向量 β 的相角 ψ 先从 π 开始，先减小（顺时针旋转）再增加，在 $\omega=\pi$ 处，ψ 恢复为 π。可见在单位圆内的零点，可以产生正相移，或者帮助减少系统的负相移。

由于稳定性的要求，系统的极点都在单位圆内，所以产生的都是负相移。由于因果性的要求，系统的极点数一定不少于零点数；因此在 $\omega=\pi$ 处系统的总相位，一定是非正数，最大为零。如果系统所有的零点都在单位圆内，系统的负相移最小，称为最小相位系统。

5.4.2　非单位圆周上的频谱分析

在很多实际应用中，并非整个单位圆上的频谱都很有意义。例如，对于窄带信号，往往只希望对信号所在的一段频带进行频谱分析，这时便希望采样能密集地在这段频带内进行，而带外部分可不予考虑。另外，有时希望采样点不局限于单位圆上。例如，语音信号处理中，常常需要知道信号极点所对应的频率，如果极点位置离单位圆较远，则其单位圆上的频谱就很平滑，如图 5.4.2（a）所示，这时很难从中识别出极点对应的频率。如果使采样点轨迹沿一条接近这些极点的弧线或圆周进行，则采样结果将会在极点对应的频率上出现明显的尖峰，如图 5.4.2（b）所示。这样就能较准确地测定出极点频率。因此就有必要计算半径为 r 的圆上的 N 点等间隔频谱。

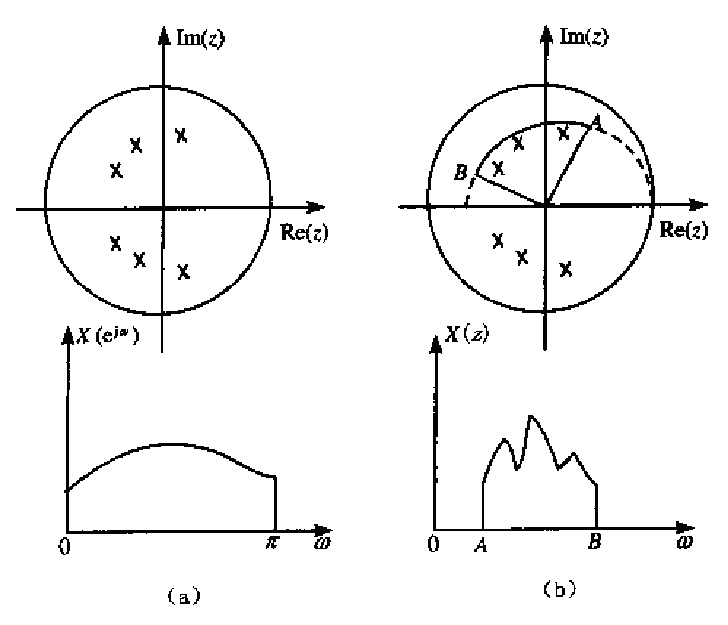


图 5.4.2　z 平面不同轨迹上极点对幅特性的影响

对均匀分布在以原点为圆心的任何圆上的 N 点（$z_k = r \cdot e^{j2\pi k/N}$，$k$=0，1，…，$N-1$）频谱采样，其结果为

$$X(z_k) = \sum_{n=0}^{\infty} x(n)(z_k)^{-n} = \sum_{n=0}^{\infty} x(n)\left(re^{j\frac{2\pi}{N}k}\right)^{-n} = \sum_{n=0}^{\infty}\left(x(n)r^{-n}\right)e^{-j\frac{2\pi}{N}kn} \tag{5.4.4}$$

上式说明，要计算 $x(n)$在半径为 r 的圆上的 N 点等间隔频谱分量，可先对 $x(n)$乘以 r^{-n}，再计算 N 点 DFT（FFT）即可得到。若要求 $x(n)$分布在该圆的有限角度内的 N 点等间隔频谱分量，可以通过尾部补零的方法，仍按式（3.5.1）用 DFT 分析整个圆上的等间隔频谱，最后只取所需角度内的频谱分量即可。

例 5.4.2：设序列信号为 $x(n) = (0.91)^n \cdot u(n)$，取 64 点，求其频谱特性，并求其在 z 平面上 $r = 0.9$ 的中心圆上的频谱。

解：此序列的 z 变换为 $X(z) = \dfrac{1}{1-0.91z^{-1}}$，即在 $z = 0.91$ 处有一个单实极点。所以 $r = 0.9$ 的圆将经过此极点附近，预计在该点附近 $H(z)$会有很大的幅特性。列出 MATLAB 程序 hc542 如下：

```
n=0:63;p=0.91; x=p.^n;        % 给定序列参数
H=freqz(x,1,64);              % 求序列的频率响应H
x1=x.*(0.9).^-n;              % 将序列乘以r^-n得到x1，
H1=freqz(x1,1,64);            % 求x1的频率响应H1等价于求x在圆r上的频率响应
绘图语句从略。
```

程序运行的结果见图 5.4.3，其中(a), (c), (e)三个子图为在单位圆上作的频谱分析；(b),(d),(f)三个子图为在半径 r 的圆周上作的频谱分析。可以看出，以 r 为半径的圆上的幅频谱子图(b),(d),(f)中，在 ω=0 处有一个冲激尖峰，另外其幅特性有不小的波动。这是因为由 x 乘以 r^{-n} 所得到的 $x1$，已经近似于常数序列，在 n=64 处突然变零，意味着有一个截断，从而引起了幅特性的波动。原来的 x 序列在 n=64 处几乎已衰减为零，所以没有截断效应的影响。

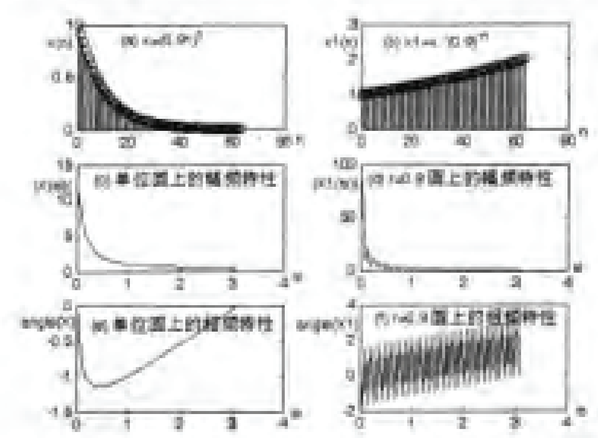


图 5.4.3　求 z 平面上任意半径 r 的中心圆上的频谱

5.4.3　z 平面螺旋线上的谱分析（Chirp-z 变换）

在 z 平面上取的另一种轨迹是而沿螺旋弧线采样，它采用线性调频 z 变换（Chirp-z 变换，简称 CZT）计算。设序列 x（n）长度为 N，要分析 z 平面上 M 点频谱采样值，其分析点为 k=0，1，…，M-1。设

$$z_k = AW^{-k} \qquad 0 \leq k \leq M-1 \tag{5.4.5}$$

式中 A 和 W 为复数，用极坐标形式表示为

$$\left.\begin{aligned} A &= A_0 e^{j\theta_0} \\ W &= W_0 e^{-j\varphi_0} \end{aligned}\right\} \tag{5.4.6}$$

式中，A_0 和 W_0 为实数。当 k=0 时，有

$$z_0 = A_0 e^{-j\theta_0} \tag{5.4.7}$$

由此可见，(5.4.5)式中，A 决定频谱分析起始点 z_0 的位置，W_0 的值决定分析路径的螺旋趋势，φ_0 则表示两个相邻分析点之间的夹角。如果 W_0<1 ，则随着 k 增大，分析点 z_k 以 φ_0 为步长向外盘旋；而 W_0>1 时，向内盘旋，如图 5.4.4 所示。

将 z_k 代入 z 变换公式得到

$$X(z_k) = \sum_{n=0}^{N-1} x(n)[AW^{-k}]^{-n} = \sum_{n=0}^{N-1} x(n)A^{-n}W^{-kn} \qquad 0 \leqslant k \leqslant M-1 \tag{5.4.8}$$

利用关系式 $nk = \frac{1}{2}[n^2 + k^2 - (k-n)^2]$ 得到

$$X(z_k) = \sum_{n=0}^{N-1} x(n)A^{-n}W^{-[n^2+k^2-(k-n)^2]/2} = W^{k^2/2}\sum_{n=0}^{N-1} x(n)A^{-n}W^{-n^2/2}W^{-(k-n)^2]/2}$$

令

$$y(n) = x(n)A^{-n}W^{n^2/2} \tag{5.4.9}$$

$$h(n) = W^{-n^2/2} \tag{5.4.10}$$

最后得到

$$X(z_k) = W^{k^2/2}\sum_{n=0}^{N-1} y(n)h(k-n) \qquad 0 \leqslant k \leqslant M-1 \tag{5.4.11}$$

(5.4.11)式说明，长度为 N 的序列 x（n）的 M 点 Chirp-z 频谱分析可通过 y（n）与 h（n）的卷积求得，而 y（n）与 h（n）分别由公式(5.4.9)与(5.4.10)算出。计算过程的示意见图 5.4.5。

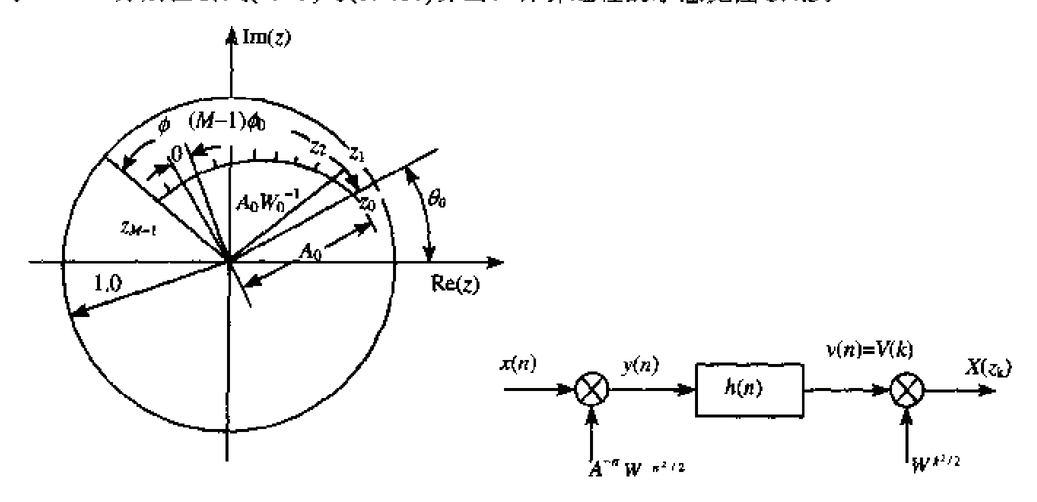


图 5.4.4　螺旋轨迹的参数　　　　图 5.4.5　Chirp-z 的计算框图

当 W_0=1 时，$h(n) = e^{-jn^2\varphi_0/2} = e^{-j(\omega_0 n)n}$，可以想像为其频率 $\omega=\omega_0 n$ 随时间线性增长的复指数序列。在雷达系统中这样的信号称做线性调频信号，并用词汇 Chirp 表示。因此对上述变换起名为线性调频 z 变换，简称 Chirp-z 变换（CZT）。

按(5.4..9) ~(5.4..11)式用手工计算 Chirp-z 变换是很难想像的。MATLAB 信号处理工具箱设有线性调频 z 变换函数 czt.m。其调用方法是 G=czt((x,M,W,A)。

其中：x 为输入序列；

M 为所取 czt 变换的点数,可以取 x 的长度，若取其他值时，x 会自动截断或补零；

W 为相邻两频点的比，是一个复数，$W = W_0 e^{-j\varphi_0}$，其模 W_0 大于 1 时，螺旋内转；

A 为起始频点，也是一个复数。

下面举例说明它的应用方法。

例 5.4.3：设序列信号为 $x(n) = (0.9)^n \bullet [\cos(\frac{\pi}{3}n) - 2\times0.2887\sin(\frac{\pi}{3}n)]$，$n\geqslant0$。取 64 点，求其频谱及在 z 平面上 $r = 0.9$ 的中心圆上的频谱，并求它在一根通过 1+0i 和 0.4500 + 0.7794i 两点的 z 平面螺旋线上的频谱。

解：此序列的 z 反变换为 $X(z) = \dfrac{1}{1-0.9z^{-1}+0.81z^{-2}}$，可以求出它有两个共轭极点 0.4500±0.7794i。在调用 czt 函数时，主要的难点是确定参数 W，使所取的螺旋线通过所关心的极点附近。如果点数 M 已经确定，通常螺旋线取在上半平面，故相角（即数字频率）的增量为 $\phi_0=\pi/M$。根据第一点到关心点之间的相角为 $\pi/3$，现在把 π 分成 64 步，$\pi/3$ 就对应于 Mx=64/3 步。今两点的模值比为 R=1/0.9，W_0 的取法应为

$$W_0 = \sqrt[M_x]{R} \tag{5.4.12}$$

解本题的 MATLAB 程序 hc543 可列写如下：

```
% 线性调频 z 变换幅特性计算程序 hc543
M=64,n=0:M-1;dw=pi/M;                                   % 规定点数，求频率分辨率
x=0.9.^n.*(cos(pi*n/3)+2*0.2887*sin(pi*n/3));          % 给定 x 序列
subplot(3,2,1),stem(n,x,'.')                            % 画序列曲线
H=freqz(x,1,M);                                         % 求序列频率响应
subplot(3,2,2),plot([0:M-1]*dw,abs(H))                  % 画 H 的幅频特性
x1=x.*(0.9).^-n; H1=freqz(x1,1,64);                     % 求 r=0.9 的中心圆上的 H(z)，先修改 x
subplot(3,2,3),plot([0:M-1]*dw,abs(H1))                 % 画 H1 的幅频特性
A=1;W0=0.9^(-3/M),W=W0*exp(-j*dw);                      % 确定螺旋线轨迹参数 A 和 W
z = A * W.^(-(0:M-1));                                  % 写出螺旋线轨迹方程
subplot(3,2,4),plot(z),grid on                          % 画出螺旋线轨迹
hold on,plot(p(1),'x')                                  % 把序列极点画在螺旋线轨迹图上
G=czt(x,M,W,A);                                         % 调用 czt 函数
subplot(3,2,5),plot([0:M-1]*dw,abs(G(1:M)))             % 画出 czt 函数的幅特性
subplot(3,2,6),plot([0:M/2-1]*dw,abs(G(1:M/2)))  % 画前半段 czt 的幅特性以显示峰值
```

程序运行所得到的曲线见图 5.4.6。从图中可看出，在 z 平面上取靠近序列极点的轨迹来计算幅特性，可以得到更明显的峰值曲线。子图(e)中在 ω 靠近 π 处幅特性迅速增长，掩盖了所关心的极点附近的特性，因此又画了子图(f)。画这个图所用的数据仍然和子图(e)一样，只是取了前面一半，即 0～π/2 那一段。于是 π/3 附近的极点幅特性就得到了明显的反映。

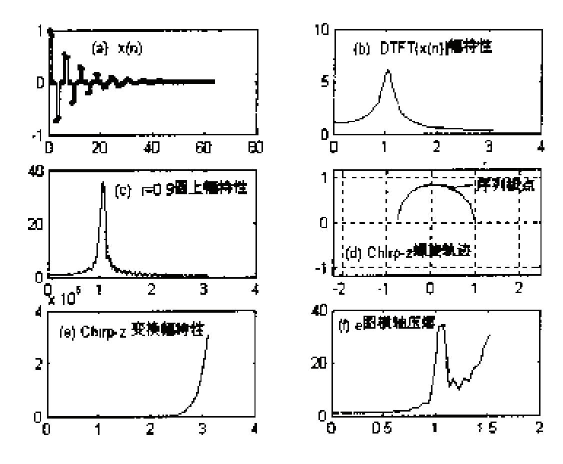


图 5.4.6　Chirp-z 变换的频谱计算及单位圆、r=0.9 圆上幅特性的比较

5.5　理想滤波器

信号处理的一个重要任务是设计处理信号的系统——滤波器。滤波器应该让需要的有用信号无失真地通过，而把不需要的信号和干扰阻挡掉。大多数情况下，有用信号和干扰信号是在不同的频段上，此时，把滤波器的频率特性进行精心的设计并实现就能达到这个目的。这种滤波器称为选频型滤波器。本书将主要讨论这种滤波器的设计方法，第 7、第 8 两章将专门讨论这个问题。在本章将着重探讨对滤波器的要求，以及一些简单的可以从零极点分布判断其特性的滤波器。目的在于使读者从基本概念上进一步加深对变换域的理解。

5.5.1　理想的因果低通滤波器的频率特性

一个理想的低通数字滤波器应该具有如下频率特性。

$$H(e^{j\omega})=\begin{cases}1\times e^{-j\omega\tau} & |\omega|\leqslant\omega_c\\ 0 & \omega_c\leqslant|\omega|\leqslant\pi\end{cases}\tag{5.5.1}$$

其曲线见图 5.5.1。从幅特性看，它可以让$|\omega|\leqslant\omega_c$的频谱分量 100%地通过，而让$|\omega|>\omega_c$的频谱分量 100%地衰减。从相特性看，它使通过的信号产生了一个相位迟延$-\omega\times\tau$，τ 是一个正的常数，在数字滤波器中，τ 的量纲不是时间，而是无量纲的拍数，所以往往用 n_0 来表示。因此这个相位迟延是与数字频率 ω 成线性关系的，也称为线性相位。

迟延也可以看成是一种失真。为什么容许一个线性相位迟延呢？如果让 $\tau=0$,对所有通过的频率都没有相移，那似乎更好，但下面可以看到，这种滤波器的脉冲响应 $h(n)$是一个双向序列，因而是不可实现的非因果系统。从物理上也不难想像，让信号通过任何一个实际的、对它进行处理的系统，都需要处理时间,这点迟延通常是可以容忍的，重要的是不要有波形的失真。如果让所有的频谱分量都迟延同样的时间，那么整个波形将原封不动地保持它的形状，仅仅迟延了一定的时间而已，因为相位＝频率×时延。

如果迟延不变，那么由于频率不同造成的相位迟延就必须与频率成线性关系。考虑到因果性，迟延不可能是正的，所以相位也必须是负的。这个问题实际上在定义 z 算子时已经隐含地提出了。$z^{-1}=e^{-j\omega}$在时域表示迟延一拍，也在频域表示了一个与 ω 成线性关系的负相移。在连续系统中，一个迟延时间为 τ 的纯迟延环节的拉普拉斯变换为$e^{-s\tau}$。这些都前后一致地说明了迟延和线性负相移的对应关系。

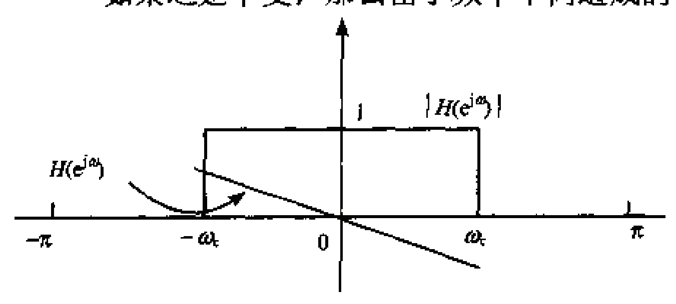


图 5.5.1　理想因果低通滤波器的幅特性和相特性

假如将一个正弦序列 $x(n)=\sin(\omega_0 n)$加到此滤波器的输入端，则经过一段时间，暂态过程消失后的稳态输出为

$$y_{ss}(n)=\left|H(e^{j\omega_0})\right|\sin[n\omega_0+\angle H(e^{j\omega_0})]\tag{5.5.2}$$

即

$$y_{ss}(n)=\begin{cases}|H(e^{j\omega_0})|\sin[\omega_0(n-n_0)]=\sin[\omega_0(n-n_0)] & |\omega_0|\leqslant\omega_c\\ 0 & |\omega_0|>\omega_c\end{cases}\tag{5.5.3}$$

最后，在通带内的信号会 1:1 无失真地通过，但延迟了 n_0 拍，而通带外的信号则完全衰减。再来看看这个理想滤波器的脉冲响应，用傅里叶反变换求得

$$h_d(n)=\frac{1}{2\pi}\int_{\omega=-\pi}^{\pi}H(e^{j\omega})e^{jn\omega}d\omega=\frac{1}{2\pi}\int_{\omega=-\omega_c}^{\omega_c}e^{-jn_0\omega}e^{jn\omega}d\omega=\frac{\sin[(n-n_0)\omega_c]}{(n-n_0)\pi} \quad (5.5.4)$$

令 $\omega_c=1$，$n_0=6$，$h_d(n)$可以用下列语句方便地画出：

```
wc=1;n0=6;n=-[10:30]+1e-5;
        % 使n偏移，避免n=n0处的不定式
h=sin(n-n0)./(n-n0)/pi;stem(n,h)
        % 绘图
```

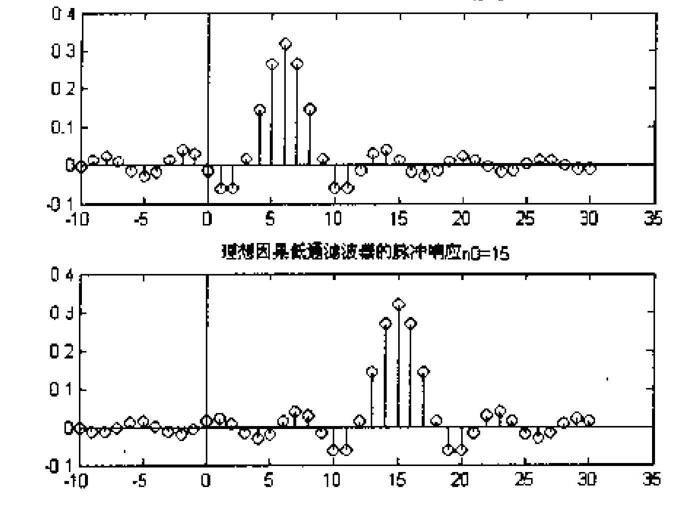


图 5.5.2　理想低通滤波器的脉冲响应，n_0与因果性的关系

得到的脉冲响应见图 5.5.2(a)，此脉冲响应是一个双向序列。考虑到因果性，只能截取 $n\geqslant 0$ 的部分,而考虑到线性相位的要求，截取的脉冲序列必须对称(下节将给予证明)。图 5.5.2(a)中 n_0=6，截取的长度只能取 13。这必然使它的频谱形状较大地偏离理想形状。为了使它的幅频特性尽量与理想情况接近，只有增加 n_0。设 n_0=15，得出的脉冲响应见图 5.5.3(b)，这样截取出 $n\geqslant 0$ 的部分就达到 31，从而使滤波器的频率特性更接近理想滤波器，付出的代价则是时延增长到 15 拍。

除了低通滤波器之外，还有高通滤波器、带通滤波器和带阻滤波器。各类滤波器的理想频率特性见图 5.5.3，也可以求出它们的脉冲响应。它们的频率特性和脉冲响应都可以由低通滤波器叠合而成。

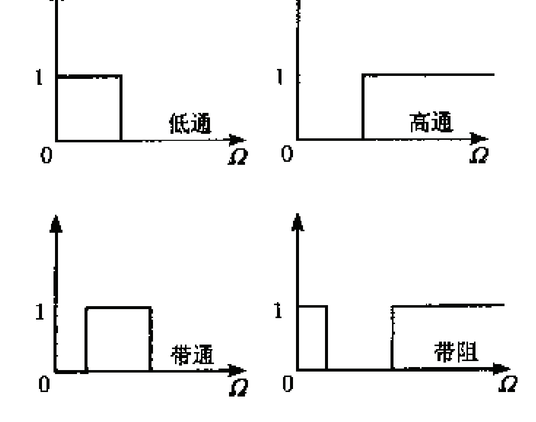


图 5.5.3　几种理想滤波器的幅频特性

5.5.2　线性相位条件对脉冲响应对称性的要求

如果求出上述截取出的因果脉冲响应的 DTFT，可以得到该滤波器实际的频率特性。那么应该向右截取多长？是不是越长越好？为了保证线性相位条件，截取的部分必须满足左右对称的规则。比如 n_0=15 作为对称中心，则 $h(14)=h(16),\cdots,h(1)=h(29),h(0)=h(30)$。证明如下：

$$H(e^{j\omega})=\sum_{n=0}^{N}h(n)e^{-j\omega n}=\left|H(e^{j\omega})\right|e^{j\angle H(e^{j\omega})}$$

希望它的相角部分与 ω 成线性关系，即满足 $\angle H(e^{-j\omega})=-\omega n_0$。先任意取两项来分析。令

$$H=h_1e^{-j\omega n_1}+h_2e^{-j\omega n_2}=\left|H\right|e^{-j\omega n_3}$$

可以把这个式子看做复平面上两个向量的相加。这两个向量的模分别为 h_1 和 h_2，而其相角分别为 $\theta_1=-\omega n_1$ 和 $\theta_2=-\omega n_2$，设合成向量的模为 h_3，相角为 θ_3。这其实就是一个简单的三角关系，如图 5.5.4 所示。问题是：如果 θ_1 和 θ_2 都加 k 倍，要使 θ_3 也加 k 倍，则 h_1 和 h_2 应满足

什么关系？

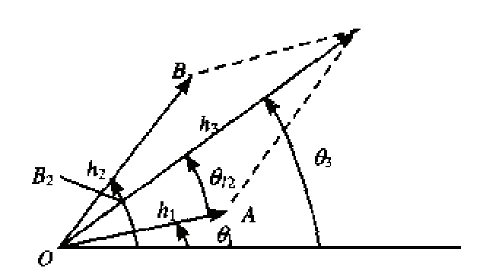


图 5.5.4　两个向量合成的三角关系

从三角关系可以看到，如果 $h_1=h_2$,则三角形 OAC 是一个等腰三角形，其两个底角相等，由此可得：

$$\theta_3=\theta_1+\theta_{12}=\theta_1+(\theta_1-\theta_2)/2=(\theta_1+\theta_2)/2=-\omega(n_1+n_2)/2=-\tau\omega$$

其中 $\tau=(n_1+n_2)/2=n_0$

可见，在这种情况下，θ_3 与 θ_1 和 θ_2 成线性关系，它的迟延量为原来两个向量迟延量的平均数。如果 $h_1\neq h_2$，θ_3 与 θ_1 和 θ_2 呈复杂的超越函数，无法构成简单的线性关系。

当 N 个向量合成，要想保证每个向量的相角增长与合成向量的相角增长满足线性关系，那就不仅要求它们的大小成对出现，而且要求各对合成的向量都在同一方向。即合成的向量都取同样的 $(n_1+n_2)/2$。这样各对合成向量之间只要幅度做代数的加减，就不会影响相角的线性关系。所以最后的结论就是，合成的向量也即 $H(\mathrm{j}\omega)$ 的相角 $-\omega n_0$ 中的 n_0 应该就是序列的对称中心点，每一对向量都应该以它为中心对称配置。

当 N 是奇数时，序列位置从 $n=0$ 到 $n=N-1$，对称条件应为 $h(0)=h(N-1)$，$h(1)=h(N-2)\cdots$。对称中心在 $n_0=(N-1)/2$ 为整数处，该处本身有一个样本点。

如果 N 是偶数，对称条件同上。而对称中心位于两个样本点的中间，这时 $n_0=(N-1)/2$ 不是整数，而是整数之半，上述线性相位的论证仍然有效。

在反对称的情况下，$h_1=-h_2$,上述几何关系仍正确，所以相应的论证也成立，它相当于把 h_2 的相角加了一个 π，从而 $\theta_3=-\omega(n_1+n_2)/2-\pi/2=-\pi/2-\tau\omega$。所以反对称的序列构成的滤波器，其相位中除了线性项 $-\tau\omega$ 外，还有一个等于 $-\pi/2$ 的常数项。同样它也可以根据 N 为奇数或偶数，分成两类。

当 N 是奇数时，序列位置从 $n=0$ 到 $n=N-1$，反对称条件应为 $h(0)=-h(N-1)$, $h(1)=-h(N-1)\ \ldots$, $h(n)=-h(N-n),\ldots$。对称中心在 $n_0=(N-1)/2$ 为整数处，该处本身有一个样本点。

如果 N 是偶数，反对称条件同上。而对称中心位于两个样本点的中间，这时 $n_0=(N-1)/2$ 不是整数，上述线性相位的论证仍然有效。

归纳起来，能够保证线性相位的滤波器，其脉冲响应只有以下四种类型。

类型 I：N 奇数，脉冲响应对称；

类型 II：N 偶数，脉冲响应对称；

类型 III：N 奇数，脉冲响应反对称；

类型 IV：N 偶数，脉冲响应反对称。

这四种类型的滤波器用途各有不同，设计时也各有特点，在第 7 章中将深入讨论。

5.5.3　考虑因果性和对称性对理想滤波器的影响

图 5.5.2 的理想因果滤波器的脉冲响应，按线性相位条件截取，所选的序列为类型 I。现分别对 n_0=6,15 计算其频率响应：

```
n0=?;N=2*n0+1;n=[0:N-1]+1e-5;
h=sin(n-n0)/(n-n0)/pi;
[H,w]=freqz(h,1);
```

在 n0=6 时，使 N=2*n0+2，故意造成不对称情况，也做同样计算。然后用绘图语句

plot(w,abs(H)); 和 plot(w,angle(H)); 分别画出三种情况的幅特性和相特性，此程序名为 hc552。运行结果见图 5.5.5。从图中可以看出：

（1）截取的脉冲序列越短，滤波器的幅频特性与理想滤波器差别越大。

（2）截取的序列若是对称的，其相频特性呈线性，即相位随频率线性变化。若不对称，其相频特性呈明显的非线性。

需要说明的是，MATLAB 在计算相位时，采用了[-π ～π]作为 arctan 的主角范围，而且由于取点密度在高频段不足，plot 命令在连接陡峭斜线时又有缺陷，因此在高频区的相位特性形状与理论有出入。读者对此不必深究。

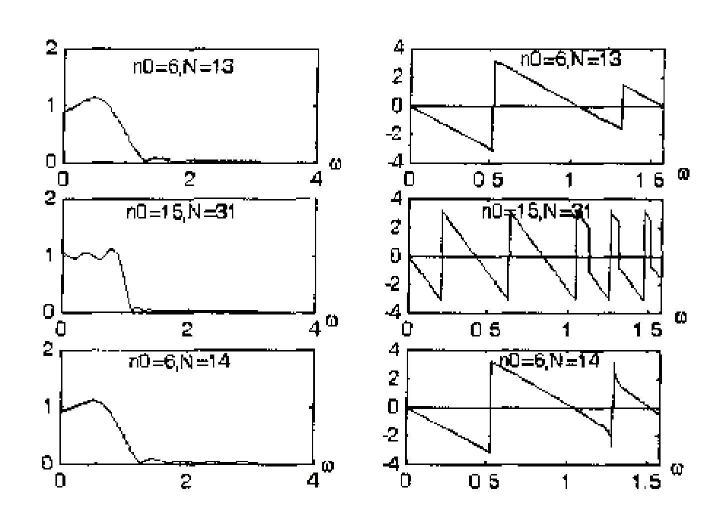


图 5.5.5　截取脉冲序列的长度及不对称性对理想滤波器特性的影响

5.6　低阶数字滤波器

低阶数字滤波器一般指的是一阶或二阶滤波器。因为传递函数的阶次较低，可以用零极点的分布来讨论它的频率特性，有比较鲜明的几何概念和物理概念。阶次低的好处是处理速度快，硬件简单，所以在高频率数字系统中应用广泛。另外，即使是高阶的滤波器，许多情况下，为了避免参数量化的影响，或者为了调整参数的方便等其他目的，往往也把它分解为多个低阶滤波器的组合来实现。掌握低阶数字滤波器的一些特点，有助于更好地理解复杂的滤波器。

5.6.1　一阶数字滤波器

一阶数字滤波器的传递函数具有如下形式：

$$H(z)=\frac{z-b}{z-a}=\frac{1-bz^{-1}}{1-az^{-1}} \tag{5.6.1}$$

它在 $z=b$ 处有一个零点，在 $z=a$ 处有一个极点。为了保证稳定性，极点应该位于单位圆之内，因而$|a|<1$。至于零点，可以在单位圆内，也可以在单位圆外。本节讨论最小相位系统，即零点在单位圆内的情况。取单位圆上的一点 $e^{j\omega}$，从极点和零点向该点引向量 α 和 β。则按图 5.6.1(a)，有 $\alpha=e^{j\omega}-a=pe^{j\phi}$ 及 $\beta=e^{j\omega}-b=qe^{j\psi}$。因而系统的频率特性为

$$H(e^{j\omega})=\frac{e^{j\omega}-b}{e^{j\omega}-a}=\frac{q}{p}e^{j(\psi-\phi)} \tag{5.6.2}$$

先设极点在原点，a=0，于是 p 为常数 1，若零点为 b= -1 时，从 ω=0 增加到 π 时，q 将从 2 变到 0， $H(\omega)$的模也将同样变化，这意味着它是一个滤波作用很弱的低通滤波器，见图 5.6.1(b)；若零点放在 b= 1 处，它就成为一个高通滤波器。总之离零点远的频率，其幅度就大，这是一条规则。对于极点，规则相反，离极点近的频率，其幅度就大。

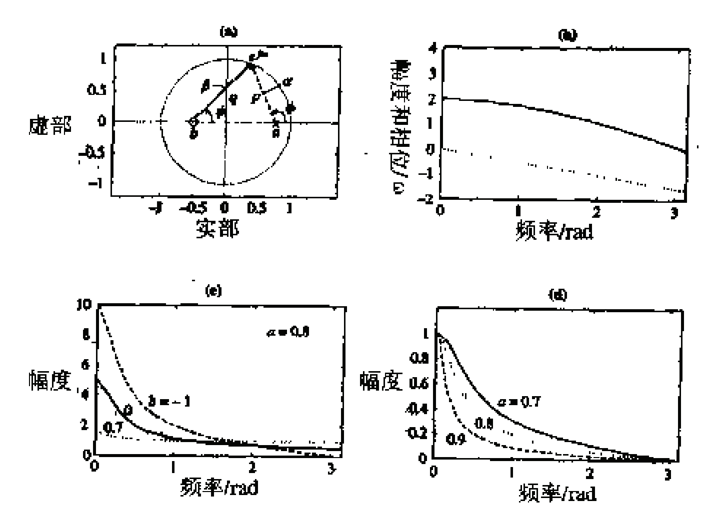


图 5.6.1 一阶滤波器零极点位置对频率特性的影响

再设极点在 a=0.8 处，零点在原点 b=0，这时 q=1 为常数，所以零点对幅特性没有影响。于是系统的频率特性将与 p 成反比。$\omega=0$ 的频点离极点最近，所以该处的幅频特性应该最大。当从 $\omega=0$ 增加到 π 时，p 将从 0.2 变到 1.8，$H(\omega)$的模也将从 5 变到 0.55。它的幅频特性见图 5.6.1(c)中的实线。改变 b 为 −1，可以得到更高的低频幅特性。而把 b 取为 0.7,和极点靠得很近，将抵消极点的影响，使得滤波器的幅特性变平，滤波效果变差了。

又设零点固定在 b=-1 处，而让 a 在 0 与 1 之间变动，研究此滤波器的通带如何变化。通带有不同的定义方法，最常用的是 3 dB 通带。dB 的定义是：

$$20\log_{10}\left|H(\mathrm{e}^{\mathrm{j}\omega})\right| = 10\log_{10}\left|H(\mathrm{e}^{\mathrm{j}\omega})\right|^2 \tag{5.6.3}$$

设滤波器在零频率处的幅特性为 1（0 dB），把它的幅特性降到-3 dB 时的频率称为 3dB 通带。为了把（5.6.1）式的零频率增益调整为 1，将它乘以一个常数 b_0=(1-a)/2，于是（5.6.1）式成为（5.6.4）式。

$$H(z) = \frac{b_0(z+1)}{z-a} = \frac{1-a}{2}\frac{z+1}{z-a} \tag{5.6.4}$$

它满足 $H(\mathrm{e}^{\mathrm{j}\omega})\Big|_{\omega=0} = H(\mathrm{e}^{\mathrm{j}0}) = H(1) = 1$ 。在 a 取 0.7, 0.8, 0.9 三种数值时，它的频率特性见图 5.6.1(d)。可以计算出它的通带 ω_p 近似公式为（5.6.5）式，这也就是极点 a 离开单位圆的最小距离。

$$\omega_{\mathrm{p}} \approx 1-a \tag{5.6.5}$$

例 5.6.1：设有一连续信号 $x(t)=\sin 10t+\cos 300t$，要求设计一阶数字低通滤波器，即滤掉高频分量而保留低频分量。

解：连续信号可以表为指数形式 $x(t)=\frac{1}{2}(\mathrm{e}^{\mathrm{j}10t}-\mathrm{e}^{-\mathrm{j}10t})+\frac{1}{2}(\mathrm{e}^{\mathrm{j}300t}+\mathrm{e}^{-\mathrm{j}300t})$，因此它的频谱是位于 Ω={-300, -10,10,300}（单位为 rad / s）四个频点上的冲击函数，其权重均为 π 。为了使数字序列能够反映信号的最高频率，采样周期 T 必须小于 π/300=0.0105s，假如取 T=0.005s。将模拟频率 10 和 300 乘以采样周期 T=0.005，得到这两个频谱分量的数字频率分别为 0.05 rad/s 和 1.5 rad/s 。因此滤波器的通带取 0.1 rad / s 就够了。从（5.6.5）式算出，a=1 - ω_p =0.9，于是取数字滤波器的传递函数为 $H(z)=\frac{1-0.9}{2}\frac{z+1}{z-0.9}=\frac{z+1}{20z-18}$。

本题信号经采样后通过此滤波器后的输出，可用下列程序 hc561 计算：

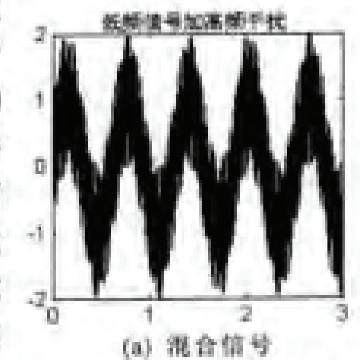
```
t=0:0.005:3; x1=sin(10*t);              % 原始的经采样后的低频信号x1
x=x1+sin(300*t);                        % 低频信号x1加高频干扰
b=[1 1];a=[20 -18];                     % 滤波器系数
y=filter(b,a,x);                        % 求滤波器输出y
subplot(1,2,1),plot(t,x)                % 画原始合成信号
subplot(1,2,2),plot(t,y,t,x1,':')       % 画原低频信号x1及输出信号y
[H,w]=freqz(b,a,[0.05,0.15,1.5])        % 求出三个频点的频率特性
abs(H), angle(H)*180/pi                 % 显示三个频点的幅度和以度（°）为单位的相角
```

程序运行所得的波形见图 5.6.2。在 ω=[0.05,0.15,1.5](即 Ω=[10,30,300])处，滤波器的频率特性的数据结果如表 5.6.1 所示。

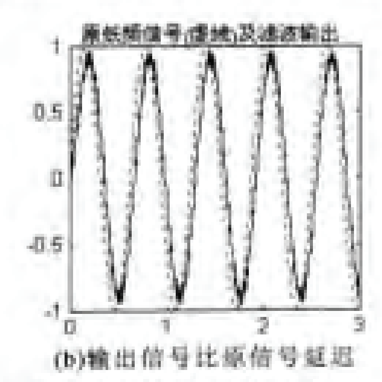
表 5.6.1　滤波器的频率特性数据

频率 Ω / rad / s	10	30	300
幅特性 $\lvert H(\Omega)\rvert$[1]	0.9032	0.5737	0.0564
相特性 / °	-25.4123	-54.9911	-86.7664

从结果中看到，输出信号中的高频分量已经衰减到约 5%，低频信号则保留了 90%，而输出低频信号比输入的低频信号（虚线）有-25.4° 的迟延。在本例中由于有用信号与干扰信号的频率差了 30 倍，因此用简单的一阶滤波器还能将有用信号和干扰加以区分。当有用信号与干扰信号的频带比较接近时，比如干扰频率为 30rad/s ，这个一阶滤波器只能滤掉 43%，滤波作用是很差的，此时就必须要用精心设计的高阶滤波器。



(a) 混合信号



(b)输出信号比原信号延迟

图 5.6.2　混合信号通过一阶低通数字滤波器的结果

5.6.2　二阶数字滤波器

二阶数字滤波器有两个极点，它的零点可以有两个、一个、甚至没有（实际上它们在无穷远处）。假定它有两个零点和两个极点，都是复数共轭的，其传递函数公式为

$$H(z)=\frac{(z-b_1e^{j\theta_2})(z-b_1e^{-j\theta_2})}{(z-a_1e^{j\alpha_2})(z-a_1e^{-j\alpha_2})}=\frac{q_1e^{j\psi_1}q_2e^{j\psi_2}}{p_1e^{j\phi_1}p_2e^{j\phi_2}} \tag{5.6.6}$$

按图 5.6.4 中的标注，不难看出它的幅特性和相特性的几何意义如（5.6.7）式

$$\left|H(e^{j\omega})\right|=\frac{q_1q_2}{p_1p_2},\quad \angle H(e^{j\omega})=\psi_1+\psi_2-\phi_1-\phi_2 \tag{5.6.7}$$

[1] 设零频率处的幅特性为 1。

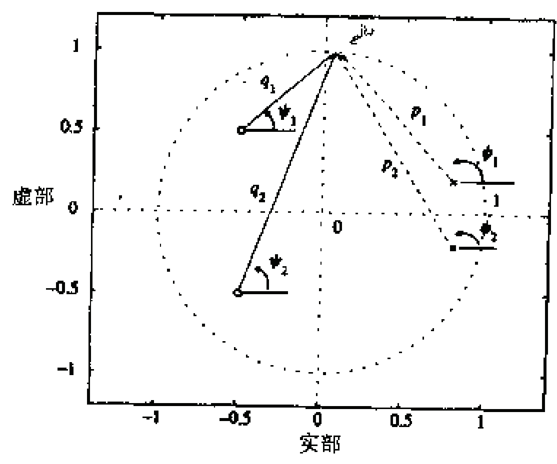


图 5.6.3　二阶滤波器的零极点分布图

从式（5.6.7）可知，q_1，q_2 越大或 p_1,p_2 越小，系统的幅特性就越高。特别是当极点很靠近单位圆时，如图 5.6.3 中的共轭极点，由于离单位圆很近，在 $\omega=\omega_0$ 处的 p_1 很小，会造成在该处很高的谐振峰。再看零点位置的影响，图 5.6.4(a)中是两个共轭复数零点，它们离开 $\omega=0$ 和 $\omega=\pi$ 处都有一定距离，因此滤波器在 $\omega=0$ 和 $\omega=\pi$ 处的幅特性就不会变成零，而且可以判断出在 $\omega=0$ 处的幅特性比 $\omega=\pi$ 处大，如图 5.6.4(b)。而图 5.6.4(c)的两个零点分别放在 $\omega=0$ 和 $\omega=\pi$ 处，该处的幅特性就会变成零，如图 5.6.4(d)所示。由于在特定的频点有特别高的谐振峰，这样的数字滤波器称为数字谐振器，可以用来选择特定频率的信号。

采用二阶滤波器来实现例 5.6.1 规定的滤波任务，比用一阶滤波器有了更大的灵活性。即使高频分量角频率为 30rad / s，数字频率为 0.15，效果也会比一阶的好很多。比如可以把两个零点配置在这个角频率的单位圆上。设两个极点都放在上次一阶极点 a=0.9 处。

$$H(z)=k\frac{(z-\mathrm{e}^{0.15\mathrm{j}})(z-\mathrm{e}^{-0.15\mathrm{j}})}{(z-0.9)(z-0.9)}=k\frac{z^2-1.9775z+1}{z^2-1.8z+0.81}=k\frac{1-1.9775z^{-1}+z^{-2}}{1-1.8z^{-1}+0.81z^{-2}}\tag{5.6.8}$$

k 是待定的增益，选择它的值使系统函数的静态增益 $H(\mathrm{e}^{\mathrm{j}0})=H(1)=1$。取

k= (1-1.8+0.81)/(1-1.9775+1)=0.4444

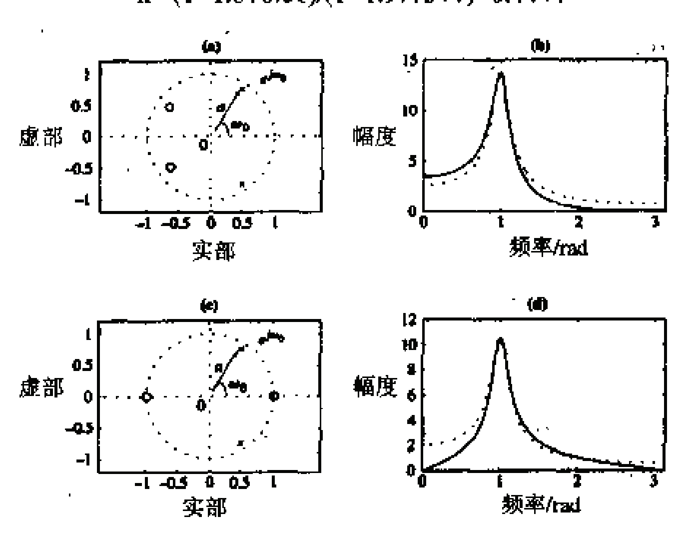


图 5.6.4　二阶滤波器零点分布与幅特性关系

此时可用下列程序求出滤波器的频率特性。

```
k=0.4444;b=k*[1,-1.9775,1];a=[1,-1.8,0.81];     % 给出滤波器参数
[H,w]=freqz(b,a); plot(w,abs(H))                 % 给出正频率范围全部幅特性
H1= freqz(b,a,[0.05,0.15,1.5]); abs(H1)          % 给出三个频率处的幅特性
```

程序运行得出幅特性如图 5.6.6 所示，数值结果为 abs(*H*1)=[1.6328　0.0014　1.0911]。从图上看出，幅特性在 $\Omega=30(\omega=0.15)$处有一个陷阱，该处的幅特性为 0.0014，使干扰减小到约 0.1%。这种滤波器称为陷波器。它通常用于滤除特定频率的干扰，比如 50Hz 的市电经常会成为高灵敏度电子仪器的主要干扰源，往往需要用陷波器来予以抑制。

图 5.6.5 二阶陷波器的幅特性

5.6.3　梳状滤波器的零极点分布

由上例可以看到，把滤波器的零点配置在单位圆上，可以产生陷波的效果，即阻挡特定频率的信号。用这个原理，把若干个零点沿单位圆按相角 0~2π 均匀分布，就可以产生梳状幅特性，这种滤波器称为梳状滤波器。它的传递函数分子具有 z^M-1 或 z^M+1 的形式。因为

$$z^M-a=0 \tag{5.6.9}$$

的 M 个根均匀分布在半径为 $a^{1/M}$ 的圆上。其值分别为

$$z_k=a^{1/M}\mathrm{e}^{\mathrm{j}2k\pi/M}\quad k=0,1,\ \cdots,M-1 \tag{5.6.10}$$

而 $z^M+a=0$ 的根则位于下列 z_k 处。

$$z_k=a^{1/M}\mathrm{e}^{\mathrm{j}(2k+1)\pi/M}\quad k=0,1,\ \cdots,M-1 \tag{5.6.11}$$

设系统的传递函数为

$$H(z)=\frac{z^4+1}{z^4+0.8} \tag{5.6.12}$$

它将有四个单位圆内的极点和四个单位圆上的零点。其零极点分布和频率特性可用下列 MATLAB 语句求得：

```
zplane([1,0,0,0,1],[1,0,0,0,0.8])
freqz([1,0,0,0,1],[1,0,0,0,0.8] ,'whole')
```

得到的零极点分布图见图 5.6.6，它的零点在单位圆上。而极点在单位圆的内部，可以保证系统是稳定的；极点的位置很靠近对应的零点，其作用是使得零点所造成的陷波特性变得很窄，仅限于零点附近。稍远一些的单位圆上的频点向零点和极点所引的向量，幅度基本相同，因此其幅特性在大部分频段上接近于 1（0 dB）。图 5.6.7(a)就表示这种很像一个梳子的幅特性，故称为梳状滤波器。

将（5.6.12）式换成负号所得的传递函数 $H(z)=\dfrac{z^4-1}{z^4-0.8}$ 所对应的零极点分布和频率特性的形状作为一个思考题留给读者。

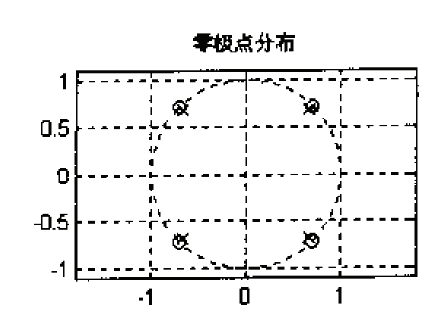


图 5.6.6 四阶梳状滤波器的零极点分布图

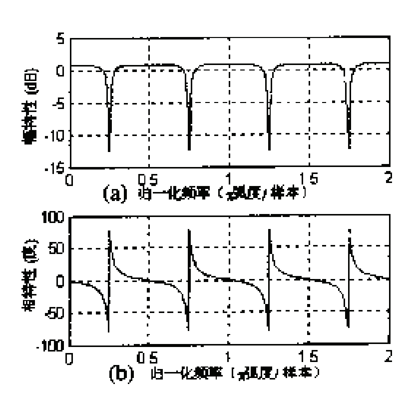


图 5.6.7 四阶梳状滤波器的频率特性

5.7　全通滤波器

如果滤波器幅频特性对所有频率均等于常数或 1，即

$$\left|H(e^{j\omega})\right| = 1, \qquad 0 \leqslant \omega \leqslant 2\pi \tag{5.7.1}$$

则称为全通滤波器。全通滤波器的频率响应函数可表示成

$$H(e^{j\omega}) = e^{j\varphi(\omega)} \tag{5.7.2}$$

（5.7.1）式表明信号通过全通滤波器后，幅度谱保持不变，仅相位谱 $\varphi(\omega)$ 随 ω 改变，起纯相位滤波作用。

全通滤波器的系统函数一般形式如下式：

$$H(z) = \frac{\sum_{k=0}^{N} a_k z^{-N+k}}{\sum_{k=0}^{N} a_k z^{-k}} = \frac{z^{-N} + a_1 z^{-N+1} + a_2 z^{-N+2} + \cdots + a_N}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}} \tag{5.7.3}$$

或者写成多个二阶滤波器级联形式：

$$H(z) = \prod_{i=1}^{L} \frac{z^{-2} + a_{1i} z^{-1} + a_{2i}}{a_{2i} z^{-2} + a_{1i} z^{-1} + 1} \tag{5.7.4}$$

上面两式中的系数均为实数，可以看出，全通滤波器系统函数 H（z）的构成特点是其分子、分母多项式的系数相同，但排列顺序相反。不难证明，（5.7.3）式表示的滤波器具有全通幅频特性。

$$H(z) = \frac{\sum_{k=0}^{N} a_k z^{-N+k}}{\sum_{k=0}^{N} a_k z^{-k}} = z^{-N} \frac{\sum_{k=0}^{N} a_k z^{k}}{\sum_{k=0}^{N} a_k z^{-k}} = z^{-N} \frac{D(z^{-1})}{D(z)} \tag{5.7.5}$$

式中 $D(z) = \sum_{k=0}^{N} a_k z^{-k}$，由于系数 a_k 是实数，所以 $D(z^{-1})\Big|_{z=e^{j\omega}} = D(e^{-j\omega}) = D^*(e^{j\omega})$。

式中 $D^*(e^{j\omega})$ 表示 $D(e^{j\omega})$ 的共轭，它们两者的模是相等的。因此有 $\left|H(e^{j\omega})\right| = \left|\dfrac{D^*(e^{j\omega})}{D(e^{j\omega})}\right| = 1$。这就证明了（5.7.3）式表示的 H（z）具有全通滤特性。

全通滤波器具有特殊的零点和极点的分布规律。设 z_k 为 H（z）的零点，按照（5.7.5）式，z_k^{-1} 必然是 H（z）的极点，记为 $p_k = z_k^{-1}$，则 $p_k z_k = 1$，即全通滤波器的极点和零点互为共轭倒易关系。如果再考虑到 D（z）和 D（z^{-1}）的系数均为实数，其极点、零点或者为实数，或者呈共轭复数对。其中，复数零点和复数极点必然以两对一组出现。例如，z_k 为 H（z）的零点，则必有零点 z_k^*，极点 $p_k = z_k^{-1}$ 和 $p_k^* = (z_k^{-1})^*$。而实数零极点则以两个一组出现，且零点与极点互为倒数关系。零极点位置的示意图如图 5.7.1 所示，$p_k = z_k^{-1}$。

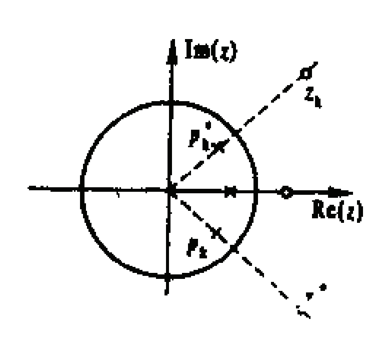


图 5.7.1　全通滤波器零极点分布

观察图 5.7.1，如果将零点 z_k 和极点 p_k^* 组成一对，将零点 z_k^* 与

极点 p_k 组成一对，那么全通滤波器的极点与零点便以共轭倒易关系出现，即 p_k 和 z_k^* 也成共轭倒易关系。因此，全通滤波器系统函数也可以写成如下形式：

$$H(z)=\prod_{k=1}^{N}\frac{z^{-1}-z_k}{1-z_k^*z^{-1}} \tag{5.7.6}$$

显然，（5.7.6）式中极点和零点互为共轭倒易关系。其全通特性的证明留给读者思考。应当注意，为了保证分子、分母多项式系数是实数，极点、零点分别以共轭对形式出现，当 $N=1$ 时，零点、极点均为实数。

全通滤波器是一种纯相位滤波器，经常用于相位均衡。对幅频特性满足要求而相频特性有缺陷的滤波器，可以级联全通滤波器进行相位校正。

复习思考题

5.1 为什么 z 变换中的收敛域是重要的?讨论收敛域时主要分清哪两类情况？

5.2 什么是左序列和右序列？为什么说工程实际中遇到的序列的 z 变换都在无穷远的邻域收敛？

5.3 求一个 z 变换的反变换时，所得的结果是不是惟一的？为什么？什么条件下求 z 反变换才能保证其惟一性？

5.4 什么是求 z 反变换的部分分式法？MATLAB 提供了什么函数帮助部分分式法？

5.5 求 z 反变换的长除法的理论基础是什么？MATLAB 提供了什么函数辅助长除法？

5.6 差分方程的解有几种分类方法？它们分类的依据各是什么？

5.7 差分方程的解的几种类别是否都可以用 z 反变换解决问题？是否都能用到部分分式法和长除法？

5.8 求零输入响应时为什么要用单边 z 变换和初值定理？怎样用法？MATLAB 提供了什么函数辅助求零输入响应？

5.9 用 z 变换如何表示系统函数，它与差分方程表示法是否等价？用框图表示系统函数时，其输入输出是否可以直接用信号序列？有什么限制？

5.10 稳态响应是什么？若加入正弦序列，则每个数字滤波器的输出也是频率相同的正弦序列，这种说法正确吗？

5.11 系统函数的负幂形式和正幂形式差别是什么？其分子分母系数向量有什么不同？在什么情况下两者的系数向量相同？

5.12 什么是系统函数的零极增益形式？为什么规定它必须取正幂形式？

5.13 数字系统的脉冲响应与它的极点位置有何关系，如果脉冲响应具有振荡衰减的特征，则该系统的极点有什么特点？

5.14 数字系统的系统函数与它的频率响应之间是什么关系？为什么说频率特性是系统函数在特定条件下的表现？为什么说系统函数能显示超越频率特性的其他系统性能？

5.15 系统的因果性和稳定性对零极点分布有什么要求？能否由零极点来判断系统的这两个特性？

5.16 能否由零极点来判断系统的频率特性？怎样估算和判断？

5.17 什么是非单位圆周上的频谱分析？什么是 Chirp-z 变换？

5.18 理想滤波器应具有什么特点？带通、高通、带阻滤波器的理想幅特性是怎样的？

5.19 为什么理想低通滤波器是无法实现的？因果的、线性相位的滤波器的脉冲响应应该

具有什么特征？

5.20 一阶低通滤波器的极点应具有什么特点？其零点呢？在最小相位滤波器范围内讨论。

5.21 二阶带通滤波器的极点应具有什么特点？其零点呢？在最小相位滤波器范围内讨论。

5.22 什么是梳状滤波器？什么是全通滤波器？各有什么特点？

习题

5.1 求以下序列的 z 变换，说明它的收敛域是在 0 还是在∞的邻域，并用 MATLAB 验算得出的 z 变换表达式。

(a) $x(n)=\{3,2,1,-2,-3\}$　　(b) $x(n)=-2^{-n}\mu(n-2)$

(c) $x(n)=(4/3)^n\mu(1-n)$　　(d) $x(n)=2^{-|n|}+(1/3)^{|n|}$

(e) $x(n)=(n+1)(3)^n\mu(n)$

5.2 利用z变换表和z变换的性质，求下列序列的z变换。把$X(z)$写成z^{-1}的有理函数。用MATLAB验算所得结果。分别说明其收敛域，并画出零极点图。

(a) $x(n)=2\delta(n-2)+3\mu(n-3)$

(b) $x(n)=(1/3)^n\mu(n-2)+(0.9)^{n-3}\mu(n)$

(c) $x(n)=n\sin(\pi n/3)\mu(n)+(0.9)^n\mu(n-2)$

(d) $x(n)=Ar^n\cos(\omega_0+\varphi)\mu(n-1)$　　$r=0.9$　$\omega_0=0.5\pi, \varphi=0.25\pi$

(e) $x(n)=(n-3)(\frac{1}{4})^{n-2}\cos(\frac{\pi}{2}(n-1)\mu(n)$

5.3 已知$x(n)$的 z变换是$X(z)=(1+2z^{-1})$, $|z|\neq 0$。求下列序列的z变换，并说明其收敛域。

(a) $x_1(n)=x(3-n)+x(n-3)$

(b) $x_2(n)=(1+n+n^2)x(n)$

(c) $x_3(n)=(1/2)^n x(n-2)$

(d) $x_4(n)=x(n+2)\,x(n-2)$

5.4 重做习题5.3，设$X(z)$具有 $X(z)=\dfrac{1-z^{-1}}{1+(5/6)z^{-1}+(1/6)z^{-2}}$, $|z|>\dfrac{1}{2}$ 的形式。

5.5 已知$X(z)$的反变换为$x(n)=(0.5)^n\mu(n)$。利用z变换的性质，分别计算出以下序列$x(n)$。

(a) $X_1(z)=\dfrac{z-1}{z}X(z)$　　(b) $X_2(z)=zX(z)$

(c) $X_3(z)=2X(3z)+3X(z/3)$　　(d) $X_4(z)=X(z)X(z^{-1})$

(e) $X_5(z)=z^2\dfrac{d}{dz}X(z)$

5.6 用MATLAB计算下列多项式运算的结果。

(a) $X_1(z)=(1-2z^{-1}+3z^{-2}-4z^{-3})(4+3z^{-1}-2z^{-2}+z^{-3})$

(b) $X_2(z)=(z^2-2z+3+2z^{-1}+z^{-2})(z^3-z^{-3})$

(c) $X_3(z)=(1+z^{-1}+z^{-2})^3$

(d) $X_4(z)=X_1(z)X_2(z)+X_3(z)$

(e) $X_5(z)=(z^{-1}-3z^{-3}+2z^{-5}+5z^{-7}-z^{-9})(z+3z^2+2z^{-3}+4z^{-4})$

5.7 假设X(z)如下 $X(z)=\dfrac{2+3z^{-1}}{1-z^{-1}+0.81z^{-2}}$，$|z|>0.9$。

(a) 求出 $x(n)$，使其不包含复数项。

(b) 用MATLAB计算$x(n)$的前20个点，并与（a）项的结果比较。

5.8 假设X(z)如下：

(a) $X(z)=\dfrac{1-(1/3)z^{-1}}{1-(1/4)z^{-1}}$　　$|z|>\dfrac{1}{2}$；　　(b) $X(z)=\dfrac{1-(1/3)z^{-1}}{1-(1/4)z^{-1}}$　　$|z|>\dfrac{1}{2}$。

分别用部分分式法和长除法计算 X(z)的 z 反变换（取 10 个点）。

5.9 用部分分式展开法和长除法分别求下列z反变换的前6个样本，设序列是右序列。

(a). $X_1(z)=\dfrac{1-z^{-1}-4z^{-1}+4z^{-3}}{1+z^{-1}+2z^{-2}+z^{-3}}$

(b). $X_2(z)=z/(z^3+2z^2+1.25z+0.25)$

(c). $X_3(z)=z/(z^2-0.25)^2$

5.10 用单边z变换解下面的差分方程，求$y(n)$。

$y(n)=0.5y(n-1)+0.25y(n-2)+x(n)$ $n>=0$，　　$y(-1)=1, y(-2)=2$

$x(n)=(0.8)^n u(n)$

用MATLAB解出$y(n)$的前20个点并与你的答案进行比较。

5.11 对于下列由脉冲响应描述的线性时不变系统，求：①系统函数表达式；②差分方程表达式；③零极点图；④输入$x(n)=(0.25)^n\mu(n)$时的输出$y(n)$。

(a)　$h(n)=2(1/2)^n\mu(n)$　　(b)　$h(n)=n(1/3)^n\mu(n)+(-1/4)^n\mu(n)$

(c)　$h(n)=3(0.9)^n\cos(\pi n/4+\pi/3)\mu(n+1)$　　(d)　$h(n)=n[\mu(n)-\mu(n-10)]$

(e)　$h(n)=[2-\sin(\pi n)]\mu(n)$

5.12 对于由下列系统函数描述的因果系统，求(i)脉冲响应的表达式或前六个样本；(ii)差分方程表达式；(iii)零－极点位置；(iv)输入$x(n)=3\cos(\pi n/3)\mu(n)$时的输出$y(n)$的前六个样本。

(a) $H(z)=(z+1)/(z-0.5)$

(b) $H(z)=(1+z^{-1}+z^{-2})/(1+0.5z^{-1}-0.25z^{-2})$

(c) $H(z)=(z^2-1)/(z-3)^2$

(d) $H(z)=(1-0.5z^{-1})/(1+2z^{-1})$，

(e) $H(z)=(1+z^{-1}+z^{-2})^2$

5.13 对于下列由差分方程描述的线性、因果、时不变系统，求：①脉冲响应表达式；②系统函数表达式；③零极点图；④输入$x(n)=2(0.9)^n u(n)$时的输出$y(n)$。

(a) $y(n)=[x(n)+2x(n-1)+x(n-3)]/4$

(b) $y(n)=x(n)+0.5x(n-1)-0.5y(n-1)+0.25y(n-2)$

(c) $y(n)=2x(n)+0.9y(n-1)$

(d) $y(n)=-0.45x(n)-0.4x(n-1)+x(n-2)+0.4y(n-1)+0.45y(n-2)$

(e)　$y(n)=\sum_{m=0}^{4}(0.8)^m x(n-m)-\sum_{l=0}^{4}(0.9)^l x(n-l)$

5.14 习题5.13的输出序列$y(n)$是全响应。对于该题中的每个系统，把$y(n)$分成：

①通解；②特解；③暂态响应；④稳态响应。

5.15 已知线性、因果、时不变系统用下列差分方程描述：

$y(n)$=0.9$y(n-1)$+$x(n)$+0.9$x(n-1)$

(a) 求系统函数$H(z)$及单位脉冲响应；

(b) 写出系统频率响应$H(\mathrm{e}^{\mathrm{j}\omega})$的表达式，并画出其幅频特性曲线；

(c) 设输入为$x(n)=[2,3,-1,-2]$，求系统输出$y(n)$。

5.16 已知线性、因果、时不变系统用差分方程$y(n)-2r\cdot y(n-1)\cos\theta+r^2y(n-2)=x(n)$描述，式中：$x(n)=a^n\mu(n)$，$0<a<1$；$0<r<1$，$\theta=$常数。试求系统响应$y(n)$。

5.17 下列系统函数的阶次是多少？是否为真分数？分别列出正、负幂条件下的分子分母系数向量，并画出它们的零极点分布。

(a) $H_1(z)=\dfrac{z^4}{(z+1)^2(z^2+0.5z+0.06)}$　　(b) $H_2(z)=\dfrac{1-z^{-3}}{z^{-1}-z^{-2}}$

(c) $H_3(z)=\dfrac{z^2+z+1}{z^3-1}$　　(d) $H_4(z)=\dfrac{1-z^{-2}}{1+z^{-1}}$

5.18 已知FIR滤波器的系统函数为$H(z)=0.1(1+0.9z^{-1}+2.1z^{-2}+0.9z^{-3}+z^{-4})$。求出此滤波器的单位脉冲响应$h(n)$，判断是否具有线性相位。并画出其幅频特性和相频特性。

5.19 若序列$h(n)$是因果序列，其傅里叶变换的实部为

$$H_{\mathrm{R}}(\mathrm{e}^{\mathrm{j}\omega})=\frac{1-a\cos\omega}{1+a^2-2a\cos\omega}\qquad |a|<1$$

求序列$h(n)$及其傅里叶变换$H(\mathrm{e}^{\mathrm{j}\omega})$。

5.20 已知线性、因果、时不变系统的系统函数为$H(z)=\dfrac{1-a^{-1}z^{-1}}{1-az^{-1}}$，　a为实数。

(a) 在z平面上用几何方法证明该系统是全通滤波器，即$|H(\mathrm{e}^{\mathrm{j}\omega})|=$常数。

(b) 画出其零极点分布图。问参数a应如何取值才能使系统因果并稳定？

5.21 对下列每个系统，求出其$H(\mathrm{e}^{\mathrm{j}\omega})$并画出其幅度和相位曲线。

(a) $y(n)=\sum_{m=0}^{6}x(n-m)$

(b) $y(n)=x(n)+2x(n-1)+x(n-2)-0.5y(n-1)-0.25y(n-2)$

第 6 章　数字滤波器结构

6.1　概述

前面已经讨论过，任何线性时不变集总参数离散系统都可用下列数学形式之一来表述其输入输出关系：在时域中，滤波器的功能可以用本书第 2 章中的差分方程(2.4.7)式或卷积公式(2.5.3)式来描述：

$$y(n)=\sum_{i=0}^{M}b_i x(n-i)-\sum_{k=1}^{N}a_k y(n-k) \tag{2.4.7}$$

$$y(n)=\sum_{m=-\infty}^{\infty}x(m)h(n-m)=x(n)\otimes h(n) \tag{2.5.3}$$

在变换域则可用本书第 5 章中的系统传递函数(5.3.7)式表示：

$$Y(z)=H(z)X(z)=\frac{b_0+b_1z^{-1}+\cdots+b_Nz^{-N}}{1+a_1z^{-1}+\cdots+a_Nz^{-N}}X(z) \tag{5.3.7}$$

这三种方法是等价的，从任何一个都能推导出其他两个。即使用同一类表示方法，也存在着许多等价的算法结构，例如：$H_1(z)=\dfrac{1}{1-0.3z^{-1}-0.4z^{-2}}$；$H_2(z)=\dfrac{1}{1-0.8z^{-1}}\cdot\dfrac{1}{1+0.5z^{-1}}$；$H_3(z)=\dfrac{0.6154}{1-0.8z^{-1}}+\dfrac{0.3846}{1+0.5z^{-1}}$。

不难证明，$H_1(z)=H_2(z)=H_3(z)$，但它们具有不同的算法。不同的算法会影响系统的某些实际性能。滤波器的工程实现要用计算机的硬件或软件帮助完成，这需要考虑许多的问题，如：

（1）计算的效率，即完成整个滤波所需要的乘法和加法次数；

（2）需要的存储量；

（3）滤波器系数的量化影响；

（4）运算中的舍入和截断误差、饱和和溢出。

不同的滤波器结构可以实现同样的传递函数，但不同的算法在满足上述的要求方面是有差别的，有的差别还很大。在研究 FFT 算法时已经看到过，同样的 DFT 运算，改变计算结构对提高计算效率和节省存储量能起多大的作用。不管用哪种方式，信号和滤波器系数都不可能有无限的精度。由于计算中的有限精度，可能使某些结构满足不了要求。因此有必要推导出基于各种结构的等效实现方法，研究用哪种结构可以减少由于系数量化对系统特性所造成的影响，从而设计出合乎要求的滤波器。

本章着重研究系统结构的等价变换方法。为了更好地讨论这个问题，需要用信号流图或状态方程来表示系统，并推导相关的系统函数表达式。

6.2　滤波器的框图结构及解法

6.2.1　信号流图

第 2 章图 2.2.2 的六种算法中当时已归结为四种，再去掉属于非线性运算的序列乘法，剩下三种基本算法，即倍率、加法和单位时延，这是数字系统处理信号的基本手段。三种基本运算用信号流图表示，如图 6.2.1 所示。

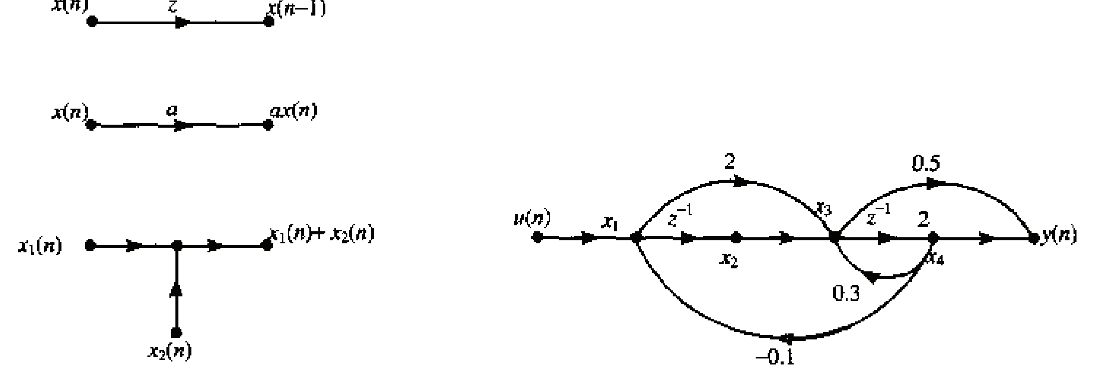


图 6.2.1　三种基本运算的流图表示　　　图 6.2.2　基本信号流图的举例

信号流图由节点和有向支路组成。每个节点表示一个信号，每个节点处的信号称节点变量。带有箭头的支路表示信号流动方向，和每个节点连接的有输入支路和输出支路，节点变量等于所有输入支路的末端信号之和。写在支路箭头旁边的 z^{-1} 或系数 a 称为支路增益。如果箭头旁边没有标明增益符号，则表示支路增益是 1。没有输入箭头的节点称为源节点或输入节点；没有输出箭头或输出箭头不指向其他节点的节点称为输出节点；两个变量相加，用指向一个圆点的两个箭头表示。因此，整个运算结构完全可用这样一些基本运算支路组成，如图 6.2.2 所示。

可见，信号流图实际上是由连接信号节点的一些有向支路构成的。按图 6.2.2 所示，可以依次对每个节点列出方程如下所示。

$$\left.\begin{aligned} x_1 &= u - 0.1x_4 \\ x_2 &= z^{-1}x_1 \\ x_3 &= x_2 + 2x_1 + 0.3x_4 \\ x_4 &= z^{-1}x_3 \\ y &= 0.5x_3 + x_4 \end{aligned}\right\} \tag{6.2.1}$$

其中有迟延算子的算式 $x_2 = z^{-1}x_1$，表示的是 $x_2(n) = x_1(n-1)$，而 $x_4=z^{-1}x_3$ 表示的是 $x_4(n) = x_3(n-1)$。从例中可以看到，用信号流图可以简明地表示系统的运算情况，同时也很容易列写它的方程。以下均用信号流图表示网络结构。

信号流图中的环路，是指信号顺着箭头流的方向能回到起点处的整条路径。有环路意味着存在反馈。上图中 x1-x2-x3-x4-x1, x1-x3-x4-x1 和 x3-x4- x3 都是环路。

不同的信号流图代表不同的运算方法，而对于同一个系统函数，可以有很多种信号流图与之相对应。从运算的可实现性考虑，信号流图必须满足以下三个条件，称为基本信号流图（primitive signal flow graphs）。

（1）信号流图中所有支路进行的都是基本运算，即支路增益是常数或者是 z^{-1}；

（2）信号流图中如果有环路，则环路中必须存在延迟支路，否则将出现信号 $x_i(n)$的计

算要依赖于 $x_i(n)$ 值的悖论状况而使数字系统无法实现。没有延迟支路的环路称为代数环，代数环是构成系统时必须避免的；

（3）节点和支路的数目是有限的。

图 6.2.2 是一个基本信号流图，图中有三个环路，环路增益分别为 $-0.2z^{-1}$，$-0.1z^{-2}$，和 $0.3z^{-1}$，说明这些环路内都有延时支路。

根据信号流图可以求出网络的系统函数，方法是列出各个节点变量方程，推导出输出与输入之间的关系。当系统内没有环路时，只要靠代入法消去中间变量就很容易解出。如果系统内存在环路，就要进行等式两端的移项合并。

例 6.2.1：求图 6.2.2 信号流图决定的系统函数 $H(z)=Y(z)/U(z)$。

解：对（6.2.1）式进行 z 变换，得到：

$$\left.\begin{aligned} &X_1(z)=U(z)-0.1X_4(z)\\ &X_2(z)=z^{-1}X_1(z)\\ &X_3(z)=X_2(z)+2X_1(z)+0.3X_4(z)\\ &X_4(z)=z^{-1}X_3(z)\\ &Y(z)=0.5X_3(z)+X_4(z)\end{aligned}\right\} \tag{6.2.2}$$

求解这个方程组纯属代数问题，但也是相当麻烦的，因为它有三个环路。现大体列出其过程，推导中把自变量符号(z)予以省略：

$$\begin{aligned} X_4 &= z^{-1}X_3 = z^{-1}(X_2+2X_1+0.3X_4) = z^{-1}((z^{-1}+2)X_1+0.3X_4)\\ &= z^{-1}\left[(z^{-1}+2)(U-0.1X_4)+0.3X_4\right] = z^{-1}(z^{-1}+2)U-(0.1z^{-2}-0.1z^{-1})X_4\end{aligned}$$

将 X_4 移项到左端，合并它的系数，并除到右端分母，可求得

$$X_4/U = z^{-1}(z^{-1}+2)/(1-0.1z^{-1}+0.1z^{-2}) \tag{6.2.3}$$

又由 $Y=0.5X_3+X_4=(0.5z^{-1}+1)X_4$　得到

$$\frac{Y(z)}{U(z)}=(0.5z+1)\cdot\frac{z^{-1}(z^{-1}+2)}{(1-0.1z^{-1}+0.1z^{-1})}=\frac{1+2.5z^{-1}+2z^{-1}}{1-0.1z^{-1}+0.1z^{-1}} \tag{6.2.4}$$

从解的过程可以看到，把方程组（6.2.1）式做 z 变换变为方程组（6.2.2）式似乎并没有得到新的信息，不过是内涵上把时域算子和信号赋予了 z 变换的含义。实际上信号流图本身是在时域建立的，时域方程中所有的推导都是正确的，而且可以实实在在地指导信号处理的实践。它惟一的问题是无法解释分母上的 z^{-1} 多项式。在掌握了 z 变换的理论并懂得 z 反变换的含义和算法以后，这个分母多项式就得到了解释。因此以后处理以信号流图为基础的滤波器结构时，不必再做一次 z 变换，可以直接用信号的时域表示式进行推导了。

6.2.2　信号流图方程的计算机解法

可以看到，当结构复杂时，上面利用节点变量方程联立求解的方法很麻烦，梅森（Masson）曾经利用网络拓扑关系提出一个公式，可以直接写出 $H(z)$ 的表示式，称为梅森公式。这在一般"信号与系统"的教科书中都已讲过。它的化简仍然是很烦琐的。利用 MATLAB 可以解决这个问题，详见参考文献[14]、[15]。

设信号流图中有 K_i 个输入节点，K 个中间节点和输出节点，它们分别代表输入信号 u_i(i=1, 2,⋯K_i）和系统状态 x_j（j=1, 2,⋯K)。信号流图代表它们之间的联结关系。在离散系统中，

这些关系由常数乘子或迟延算子 z^{-1} 构成。令 $q=z^{-1}$，则信号流图中的任何节点变量 x_j 可以表为输入变量 u_i 和其他节点变量 x_j 的线性组合：

$$x_j=\sum_{k=1}^{K}Q_{jk}x_k+\sum_{i}^{Ki}P_{ji}u_i \tag{6.2.5}$$

用矩阵表示可写成：

$$X=\begin{bmatrix}x_1\\x_2\\\vdots\\x_K\end{bmatrix}=\begin{bmatrix}Q_{11}&Q_{12}&\cdots&Q_{1K}\\Q_{21}&Q_{22}&\cdots&Q_{2K}\\\vdots&&&\vdots\\Q_{K1}&Q_{K2}&\cdots&Q_{KK}\end{bmatrix}\begin{bmatrix}x_1\\x_2\\\vdots\\x_K\end{bmatrix}+\begin{bmatrix}P_{11}&P_{12}&\cdots&P_{1ki}\\P_{21}&P_{22}&\cdots&P_{2ki}\\\vdots&&&\vdots\\P_{K1}&P_{K2}&\cdots&P_{Ki}\end{bmatrix}\begin{bmatrix}u_1\\u_2\\\vdots\\u_{ki}\end{bmatrix}=QX+PU \tag{6.2.6}$$

其中：$X=[x_1,x_2,\cdots,x_K]^T$ 为 K 阶状态变量列向量，$U=[u_1,u_2,\cdots,u_{Ki}]^T$ 为 Ki 阶输入变量列向量，Q 为 $K\times K$ 阶的连接矩阵，P 为 $K\times Ki$ 阶的输入矩阵。

把（6.2.6）式右端的 QX 项移到等式左端，有 $(I-Q)\cdot X=P\cdot U$ 。令 $W=X/U$ 为以全部 U 为输入，全部状态 X 为输出的系统函数，不难得到：

$$W=\frac{X}{U}=(I-Q)^{-1}P \tag{6.2.7}$$

其中 W 为 $K\times Ki$ 的矩阵，也称传递矩阵。若只有一个输入，则 W 为 K 阶单列向量，表示输入为 U，输出为 X(其中含 K 个变量)的 K 个传递函数组成的向量。这个简明的公式等价于信号流图中的梅森公式。只要写出 P 和 Q，任何复杂系统的传递矩阵都可用这个简单的式子求得。在没有计算工具的时候，计算$(I-Q)^{-1}$涉及一个阶次很高的矩阵的求逆运算，所以虽然简明却不好使用。但现在就不是问题了，不过还有一个关键点需要扩展。

通常计算机只能进行数值计算。已学过的 MATLAB 语言中，放在等式右边的变量，都必须是已经赋过值的。如果不给 x 赋值，键入 y=2*x，屏幕就会出现错误信息——Undefined function or variable 'x'。现在待求的传递函数中存在着一个无法赋值的符号变量 $q=z^{-1}$，显然将无法为 MATLAB 所接受。幸而 MATLAB 已经引进了符号推理的功能。只要在程序的开始，规定某些变量是符号变量，系统就不会去追究它的值，而在运算中始终保留这个符号。定义符号变量使用的命令是 syms。编写这个程序时，还有两点要说明：

（1）由于变量要分成数值变量和符号变量两类，所以变量组成的矩阵也要分成两类。一个矩阵的属性只有一个，符号变量矩阵可以有数值元素，但数值矩阵就不容许有符号变量。比如这里的 Q 矩阵，必须是符号矩阵。矩阵的属性是由第一个赋值元素的属性决定的。所以程序中给 Q 赋值的第一条语句的右端，必须含符号变量，这很重要，它保证了 Q 是符号矩阵。

（2）Q 和 P 阵都有很多个元素，但不需要逐个赋值。应该把非零的元素赋值，零元素先不用去管它，最后只要把矩阵右下角的元素赋值。利用 MATLAB 的规定，矩阵中所有左上方的未赋值元素都会自动赋为零。

现在利用计算机来解例题 6.2.1。初学者应该先列出矩阵方程：

$$X=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\end{bmatrix}=Q\cdot X+P\cdot U=\begin{bmatrix}0&0&0&0.1&0\\q&0&0&0&0\\2&1&0&0.3&0\\0&0&q&0&0\\0&0&0.5&1&0\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\end{bmatrix}+\begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix}\cdot U \tag{6.2.8}$$

由此可以得出程序 hc621 如下：

```
syms q                                  % 规定 q 为符号变量
Q(2,1)=q; Q(1,4)=-0.1;                  % 先给 Q(2,1)赋值，因为它的右端含符号变量
Q(3,1)=2;Q(3,2)=1;Q(3,4)=0.3;
Q(4,3)=q
Q(5,3)=0.5;Q(5,4)=1;
Q(5,5)=0;                               % 给 Q 右下角分量赋值，使矩阵 Q 中未赋值的分量全置零
P(1,1)=1;                               % 给矩阵 P 赋值
P(5,1)=0;                               % 给 P 右下角分量赋值，使矩阵 P 未赋值的分量全置零
W=inv(eye(5)-Q)*P                       % 按公式（6.2.7）求 X=[x1; x2; x3; x4; y] 对 U 的系统函数
pretty(W(5))                            % 以易于阅读的方式显示以 y=x5 为输出的系统函数
```

程序运行的结果为：

$$\boldsymbol{W}=\begin{bmatrix} -(3*q-10)/(10-q+q\text{^}2) \\ -q*(3*q-10)/(10-q+q\text{^}2) \\ 10*(2+q)/(10-q+q\text{^}2) \\ 10*q*(2+q)/(10-q+q\text{^}2) \\ 5*(5*q+2*q\text{^}2+2)/(10-q+q\text{^}2) \end{bmatrix} \tag{6.2.9}$$

$$W(5)=\frac{5*(2+5q+2q^2)}{10-q+q^2}=\frac{1+2.5z^{-1}+z^{-2}}{1-0.1z^{-1}+0.1z^{-2}} \tag{6.2.10}$$

（6.2.9）式中，MATLAB 把以 U 为输入，五个变量 X=[x1；x2；x3；x4；y]为输出的传递函数 W 一下全求出来了。需要的只是第五个传递函数 W(5)。用 pretty(W(5))的语句使它以易读的形式显示。因为 MATLAB 的变量不容许有上标，其显示只能到 q 为止，后面的 z^{-1} 函数形式是用人工补上去的。

6.2.3　信号流图与数字系统的网络结构图

数字滤波器的网络结构图实际上也是一种信号流图。它惟一的特殊要求在于所有的相加节点都限定为双输入相加器。如果有三个信号（x_1, x_2 和 x_3）要相加，那就要分成两个节点，x_1 和 x_2 在前一个节点相加，合成信号再在下一个节点与 x_3 相加。这样做的好处是与实际微处理器或信号处理器的硬软件结构相符合，有助于信号处理实用系统的开发。同时在结构形式上也比较规范，一目了然，便于理论分析。它的缺点是方程的数目增多，在系统函数的推导上造成一些麻烦。不过在使用计算机帮助时，这点困难是很容易克服的。

第 2 章已经指出，通常把滤波器分成两类：FIR 和 IIR，本书将分别予以介绍。

6.3　FIR 基本网络结构

6.3.1　概述

FIR 网络结构特点是没有反馈支路，即没有环路，其单位脉冲响应是有限长的。设单位脉冲响应 $h(n)$长度为 N，其系统函数 $H(z)$和差分方程分别为

$$H(z)=b_0+b_1z^{-1}+\cdots+b_{N-1}z^{1-N}=\sum_{n=0}^{N-1}b_nz^{-n} \tag{6.3.1}$$

$$y(n)=b_0x(n)+b_1x(n-1)+\cdots+b_{N-1}x(n-N+1)=\sum_{i=0}^{N-1}b_ix(n-i) \qquad (6.3.2)$$

其单位脉冲响应 $h(n)$是有限长的，按照（6.3.1）式的 z 反变换，$h(n)$表示为

$$h(n)=\begin{cases}b_n & 0\leqslant n\leqslant N-1\\ 0 & n=\text{其他}\end{cases} \qquad (6.3.3)$$

滤波器的阶数为 N-1，它的长度（等于系数的个数）为 N。FIR 滤波器总是稳定的，同 IIR 结构相比而言，相对简单一些。而且，FIR 滤波器可设计成具有线性相位，这是某些应用所希望的。现在讨论以下四种结构。

（1）直接形式：这种结构形式以直接实现差分方程（6.3.2）而得名。

（2）级联形式：它把（6.3.1）式中的多项式 $H(z)$分解成多个二阶因式，然后用级联连接。

（3）线性相位形式：当 FIR 滤波器具有线性相位响应时，根据第 5 章的 5.5 节可知，它的脉冲响应具有对称性。在这种形式下利用对称关系，能把倍率的计算量缩至一半。

（4）频率采样形式：这种结构基于脉冲响应 $h(n)$的 DFT，也可以基于频率响应的样本 $H(\omega_k)$，其结果可以形成一个并联结构。

6.3.2 直接型

设 N=5（即四阶 FIR 滤波器），按照 $H(z)$或者差分方程式画出的直接结构图如图 6.3.1 所示，它可以用抽头延迟线实现。则可以写出

$$y(n)=b_0x(n)+b_1x(n-1)+b_2x(n-2)+b_3x(n-3)+b_4x(n-4) \qquad (6.3.4)$$

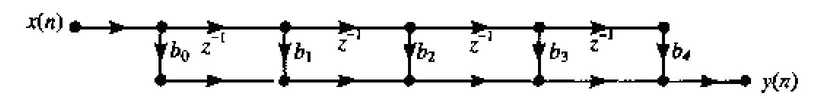


图 6.3.1 四阶 FIR 滤波器直接结构的信号流图

根据此信号流图可以直接写出方程式（6.3.4），而且不必写出联立方程了。

6.3.3 级联型

将 $H(z)$进行因式分解，并将共轭成对的零点放在一起，形成一个系数为实数的二阶形式。这样级联型网络结构是由一阶或二阶实系数因式级联构成的，其中每一个因式都可用直接型实现。

例 6.3.1：设 FIR 网络系统函数 $H(z)$如(6.5.3)式，要求画出 H（z）的直接型结构和级联型结构。

$$H(z)=0.96+2z^{-1}+2.8z^{-2}+1.5z^{-3} \qquad (6.3.5)$$

解：将 $H(z)$进行因式分解，得到：

$$H(z)=(0.6+0.5z^{-1})(1.6+2z^{-1}+3z^{-2}) \qquad (6.3.6)$$

其级联型结构和直接型结构如图 6.3.2 所示。

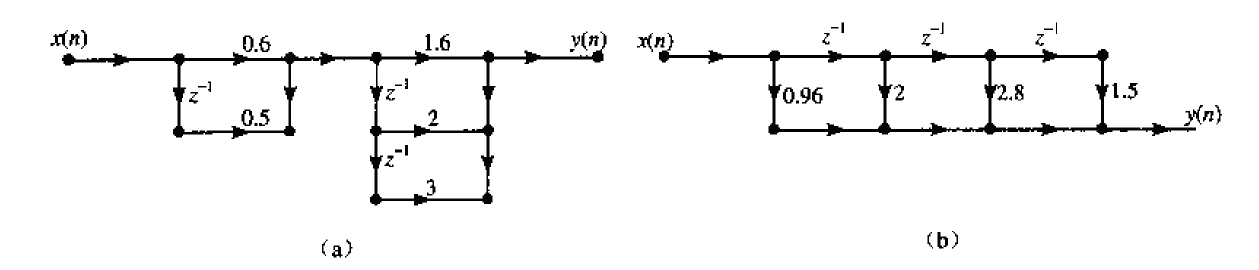


图 6.3.2　例 6.3.1 中得出的两种滤波器结构

级联型结构每个一阶因式控制一个零点，每个二阶因式控制一对共轭零点，因此调整零点位置比直接型方便，但 $H(z)$中的系数比直接型多，因而需要的乘法器多。在例 6.3.1 中直接型需要四个乘法器，而级联型则需要五个乘法器，分解的因式愈多，需要的乘法器也愈多。当 $H(z)$的阶次高于三阶时，不易分解，需要用 MATLAB 协助。

信号处理工具箱中有一个 tf2sos 函数，它的含义是 transfer function to second-order-section，即由传递函数转换为二阶环节。其调用方法为[sos,G] = tf2sos(b,a)。

其中 a,b 为系统负幂传递函数的分母、分子系数向量。

$$sos=\begin{bmatrix} b_{01} & b_{11} & b_{21} & 1 & a_{11} & a_{21} \\ b_{02} & b_{12} & b_{22} & 1 & a_{12} & a_{22} \\ & & \cdots & & & \\ b_{0L} & b_{1L} & b_{2L} & 1 & a_{1L} & a_{2L} \end{bmatrix} \tag{6.3.7}$$

其每一行代表一个二阶环节，前三项为分子系数，后三项为分母系数。对第 k 个环节,有

$$H_k(z)=\frac{b_{0k}+b_{1k}z^{-1}+b_{2k}z^{-2}}{1+a_{1k}z^{-1}+a_{2k}z^{-2}} \qquad k=1, 2, \cdots, L \tag{6.3.8}$$

G 则是整个系统归一化的增益。系统函数的最后形式应为

$$H(z)=G*H_1(z)*H_2(z)*...*H_L(z)$$

可以看出，这个函数是适用于 IIR 系统的，因为它同时把分子分母都进行了因式分解。使用于 FIR 系统时，只要把系统函数的分母系数 a 用 1 代入即可。

6.3.4　线性相位形式

第 5 章 5.5 节曾指出，理想的低通滤波器具有线性相位，如(6.3.9)式所示。

$$\angle H(e^{j\omega})=\beta-\tau\,\omega, \qquad -\pi\leqslant\omega<\pi \tag{6.3.9}$$

其中 $\beta=0$ 或$\pm\pi/2$，τ 为时延的拍数。要得到线性相位，实际滤波器的脉冲响应序列应当具有对称性。按(6.3.3)式，脉冲响应序列 $h(n)$就是滤波器分子系数向量 $b(n)$，所以系数 b 应具有对称性。利用这种对称性可以简化滤波器结构。

对于因果的 FIR 滤波器，它的脉冲响应定义在区间[0,$N-1$]上，$h(n)$和 $b(n)$有下面的对称性：

$$b(n)=b(N-n-1), \qquad \beta=0, \qquad 0\leqslant n\leqslant N-1 \tag{6.3.10}$$

$$b(n)=-b(N-n-1), \qquad \beta=\pm\pi/2, \quad 0\leqslant n\leqslant N-1 \tag{6.3.11}$$

如前所述，满足条件（6.3.10）式的脉冲响应叫做对称脉冲响应，而满足（6.3.11）式的叫做反对称脉冲响应。若差分方程（6.3.2）式具有（6.3.10）式中的对称脉冲响应。可以得到

$$y(n) = b_0x(n) + b_1x(n-1) + \cdots + b_{N-1}x(n-N+1)$$
$$= b_0[x(n) + x(n-N+1)] + b_1[x(n-1) + x(n-N+2)] + \cdots \tag{6.3.12 a}$$

图 6.3.3 表示了 N 为奇数和偶数两种情况下实现上述差分方程的方框图。

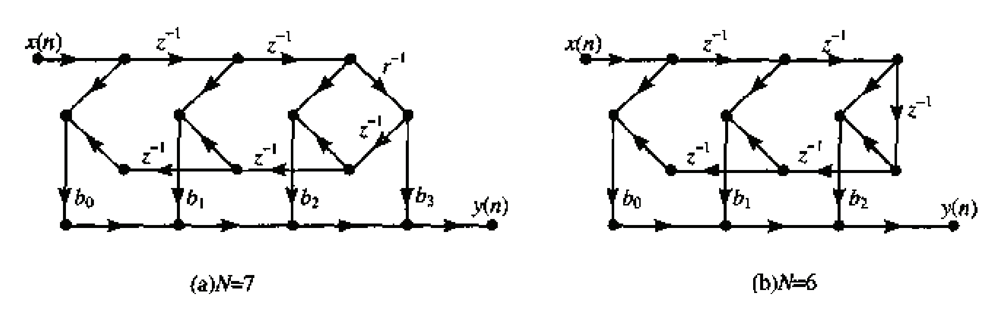


图 6.3.3　FIR 滤波器的线性相位结构（对称脉冲响应）

显然，这种结构比直接形式所需的乘法数小 50%。对反对称脉冲响应可导出类似的算法。因为 $b(n)= -b(N-n-1)$，所以把(6.3.12a)式改成(6.3.12 b)式就可以了。

$$y(n) = b_0[x(n) - x(n-N+1)] + b_1[x(n-1) - x(n-N+2)] + \cdots \tag{6.3.12 b}$$

线性相位结构在本质上仍然是直接形式，只是缩减了乘法计算量。因此，传递函数多项式的组成形式上，线性相位结构仍等于直接形式。

例 6.3.2：设 FIR 滤波器由系统函数式(6.3.13)给定，求出并画出它的直接形式、线性相位形式和级联形式的结构。

$$H(z) = 1 + 16.0625z^{-4} + z^{-8} \tag{6.3.13}$$

解：

a. **直接形式**。差分方程为式(6.3.14)，直接形式结构如图 6.3.4(a)所示。

$$y(n) = x(n) + 16.0625x(n-4) + x(n-8) \tag{6.3.14}$$

b. **线性相位形式**。差分方程可写为式(6.3.15)，所得结构如图 6.3.4(b)所示。

$$y(n) = [x(n) + x(n-8)] + 16.0625x(n-4) \tag{6.3.15}$$

c. **级联形式**。要用 MATLAB 协助

```
b=[1, 0, 0, 0, 16.0625, 0, 0, 0, 1];
[sos, G]=tf 2sos(b, 1)
```

程序运行的结果为：

$$sos = \begin{bmatrix} 1.0000 & 2.8284 & 4.0000 & 1.0000 & 0 & 0 \\ 1.0000 & -2.8284 & 4.0000 & 1.0000 & 0 & 0 \\ 1.0000 & 0.7071 & 0.2500 & 1.0000 & 0 & 0 \\ 1.0000 & -0.7071 & 0.2500 & 1.0000 & 0 & 0 \end{bmatrix}$$

G = 1

其传递函数应写成

$$H(z) = (1+2.83z^{-1}+4z^{-2})(1-2.83z^{-1}+4z^{-2})(1+0.71z^{-1}+0.25z^{-2})(1-0.71z^{-1}+0.25z^{-2})$$

级联形式结构如图 6.3.4(c)所示。

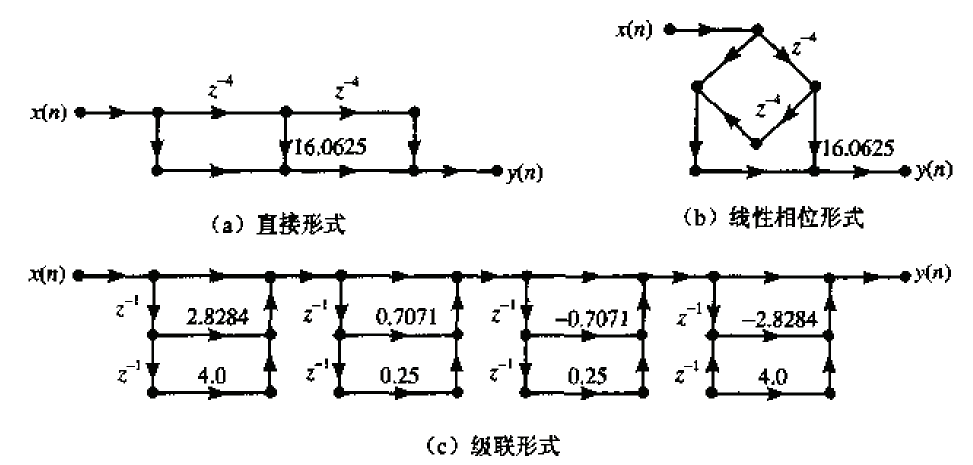


图 6.3.4　例 6.3.2 的 FIR 滤波器的三种结构形式

6.3.5　频率样本结构

由第 3 章的(3.5.11)式可知，FIR 滤波器的频率特性 $H(\mathrm{e}^{\mathrm{j}\omega})$ 可由它在单位圆上的样本 $H(k)$ 进行重构。从对 DFT 的讨论可知，这些样本实际上是 N 点脉冲响应 $h(n)$的 N 点 DFT 值 $H(k)$，$0 \leqslant k \leqslant N-1$。现在把它推广到 z 域的传递函数上，可以写出：

$$H(z)=Z[h(n)]=Z[\mathrm{IDFT}\{H(k)\}]=\sum_{n=0}^{N-1}\left[\frac{1}{N}\sum_{k=0}^{N-1}H(k)W_N^{-kn}\right]z^{-n}$$

利用 $W_N^{-kN}=1$ 及 $\sum_{n=0}^{N-1}\left(W_N^{-k}z^{-1}\right)^n=\frac{1-W_N^{-k}z^{-N}}{1-W_N^{-k}z^{-1}}$ 可以化简得到(6.3.16)式。

$$H(z)=\frac{1-z^{-N}}{N}\sum_{k=0}^{N-1}\frac{H(k)}{1-W_N^{-k}z^{-1}} \tag{6.3.16}$$

由此说明，在这种结构中，使用的是 DFT$[h(n)]=H(k)$，而不是脉冲响应 $h(n)$或者差分方程。同时也可注意到一个现象：因为(6.3.16)式包含零点和极点，由它描述的 FIR 滤波器具有与 IIR 滤波器相类似的递归形式。在 W_N^{-k} 处的极点与梳状滤波器 $1-z^{-N}$ 处的零点相抵消，因此实际上滤波器仍为 FIR 滤波器。(6.3.16)式中的系统函数是一个并联结构，图 6.3.5 给出 $N=4$ 时的结构。

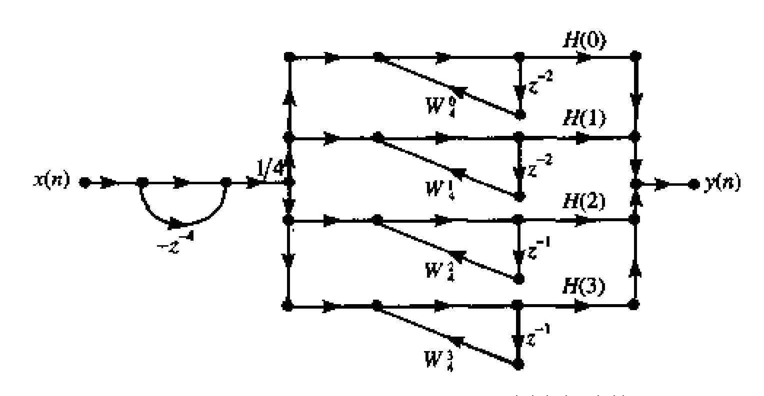


图 6.3.5　N=4 时的滤波器频率域样本结构

图 6.3.5 的频率样本结构有两个突出优点：

（1）在频率样本点 ω_k，$H(\omega_k)=H(k)$。只要调整一阶网络中乘法器的系数 $H(k)$，就可以有效地调整频响特性，在实践中相当方便。

（2）对于任何 N 阶系统的频响形状，其梳状滤波器及 N 个一阶网络部分结构完全相同，只是各支路

增益 $H(k)$不同。这样，相同部分便于标准化、模块化。

频率样本结构的缺点是：

（1）系统稳定性是靠位于单位圆上的 N 个零极点对消来保证的。实际上，存储器的字长都是有限的，这就可能使零极点不能完全对消，从而影响系统稳定性。

（2）$H(k)$和 W_N^{-k} 一般为复数，不利于硬件实现。实际滤波器总是实系数的，所以要设法将成对的共轭极点结合起来，消去虚数部分。利用 DFT 的对称性和 W_N^{-k} 因子，可把(6.3.16)式表示成(6.3.17)式。

$$H(z)=\frac{1-z^{-N}}{N}\sum_{k=1}^{L}2\left|H(k)\right|H_k(z)+\frac{H(0)}{1-z^{-1}}+\frac{H(N/2)}{1+z^{-1}} \tag{6.3.17}$$

其中：N 为奇数时，$L=(N-1)/2$；N 为偶数时，$L=N/2-1$；$H_k(z)$ 是 L 个二阶环节，它由(6.3.18)式给出。

$$H_k(z)=\frac{\cos[\angle H(z)]-z^{-1}\cos[\angle H(z)-2\pi k/N]}{1-2z^{-1}\cos(2\pi k/N)+z^{-2}} \quad k=1,2,\cdots,L \tag{6.3.18}$$

注意 DFT 样本 $H(0)$和 $H(N/2)$为实数，而且当 N 为奇数时，式(6.3.17)中的第三项不存在。根据式(6.3.17)和(6.3.18)，M=4 时含实系数的频率样本结构如图 6.3.6 所示。

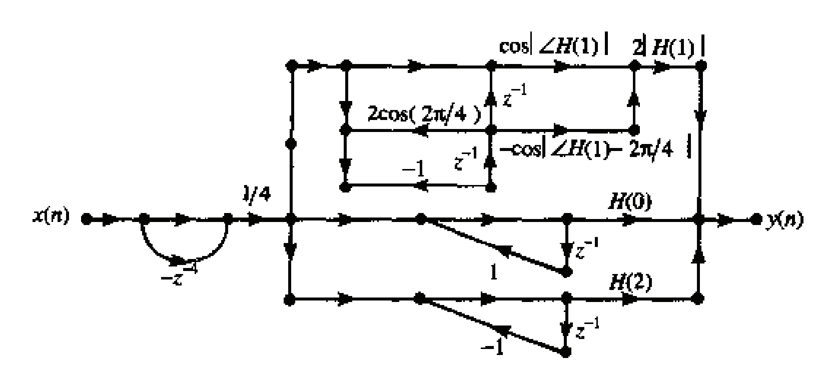


图 6.3.6　N=4 时含实系数的频率采样结构

下面讨论 MATLAB 实现。

给定脉冲响应 $h(n)$或 DFT[$h(n)$]= $H(k)$，必须求出公式(6.3.17)和(6.3.18)中的系数。本书参考文献[1]提供了由直接型结构转换为频率样本结构的函数 tf2fs。它按照公式(6.3.17)和(6.3.18)，把直接形式的 $h(n)$转换成频率样本各系数。

```
function [C,B,A] = tf2fs(h)
% C = 各并联部分增益的行向量（相当于(6.3.17)式中的|H(k)|）
% B = 按行排列的分子系数矩阵（各行相当于(6.3.18)式中的分子系数向量）
% A = 按行排列的分母系数矩阵（各行相当于(6.3.18)式中的分母系数向量）
%
N = length(h); H = fft(h,N);                   % 用FFT求出各频率样本的值
MagH = abs(H); phaH = angle(H)';               % 求出各频率样本的幅值和相位
if (N == 2*floor(N/2))                         % N为偶数
   L = N/2-1; A1 = [1,-1,0;1,1,0];             % 设置 z=±1 处两个实极点
   C1 = [real(H(1)),real(H(L+2))];             % 及相应的系数
else                                           % N 为奇数
    L = (N-1)/2; A1 = [1, -1,0];               % 设置 z=+1 处的实极点
   C1 = [real(H(1))];                          % 及相应的系数
end
k = [1:L]';
B = zeros(L,2); A = ones(L,3);                 % 初始化 B 和 A 数组
A(1:L,2) = -2*cos(2*pi*k/N); A = [A;A1];       % 计算分母系数,加上实极点系数
B(1:L,1) = cos(phaH(2:L+1));                   % 计算分子系数
B(1:L,2) = - cos(phaH(2:L+1) - (2*pi*k/N));
```

```
C = [2*MagH(2:L+1),C1]';                    % 计算增益系数
```

在上述函数中，脉冲响应值由数组 h 提供。经过变换以后，数组 C 中将包含每个并联子系统的增益值。首先给出二阶并联子系统的增益 $2|H(k)|$，然后为 $H(0)$和 $H(N/2)$（如果 N 为偶数）。矩阵 B 中为(6.3.18)式中 $H(k)$的分子系数，每个二阶环节是长度为 2 的行向量。矩阵 A 为(6.3.18)式中 $H(k)$的分母系数，每个二阶环节是长度为 3 的行向量，A 的前面各行分别与 B 各行相对应，最后多出一行或两行，它们是一阶子系统 $H(0)$和 $H(N/2)$的分母系数。

图 6.3.6 所示的结构在实际中会遇到一个问题，由于它在单位圆上存在极点，这将导致滤波器临界不稳定。为避免这个问题，对 $H(z)$在圆$|z|=r$ 上采样，其中半径 r 非常接近于 1，但比 1 小（如 $r=0.99$），得到

$$H(z)=\frac{1-r^{N}z^{-N}}{N}\sum_{k=1}^{L}\frac{H(k)}{1-rW_N^{-k}z^{-k}}H_k(z)\qquad H(k)=H(re^{j2\pi k/N})\tag{6.3.19}$$

当 $r\approx1$ 时，近似地使 $H(re^{j2\pi k/N})\approx H(e^{j2\pi k/N})$。可以得到与图 6.3.6 相似的稳定结构。

例 6.3.3：设 $h(n)=\{1/9, 2/9,3/9,2/9,1/9\}$，求出并画出频率样本结构。

解：只要直接调用上述 tf2fs 函数程序即可，其语句如下：

```
 h = [1, 2, 3, 2, 1]/9;
 [C, B, A] = tf2fs(h)
C = 0.5818
    0.0849
    1.0000
B = -0.8090    0.8090
    0.3090    -0.3090
A = 1.0000    -0.6180   1.0000
    1.0000     1.6180   1.0000
    1.0000    -1.0000   0
```

由于 $M=5$ 是奇数，因此只有一个一阶环节，因而

$$H(z)=\frac{1-z^{-5}}{5}\left[0.5818\frac{-0.809+0.809z^{-1}}{1-1.618z^{-1}+z^{-2}}+0.0848\frac{0.309-0.309z^{-1}}{1+1.618z^{-1}+z^{-2}}+\frac{1}{1-z^{-1}}\right]$$

频率样本形式如图 6.3.7 所示。由图 6.3.5 可见，当采样点数 N 很大时，显然其结构很复杂，需要的乘法器和延时单元多。但对于窄带滤波器，大部分频率样本值 $H(k)$ 为零，使二阶网络个数大大减少。所以频率样本结构适用于窄带滤波器。

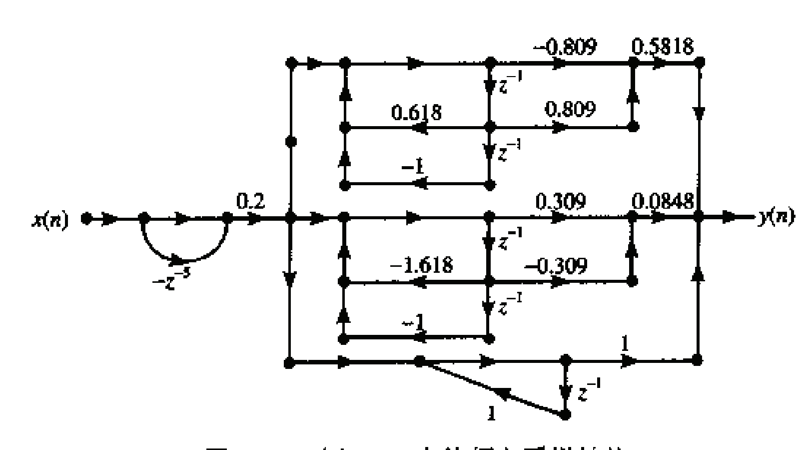


图 6.3.7　例 6.3.3 中的频率采样结构

6.4 IIR 基本网络结构

IIR 滤波器的系统函数为(6.4.1)式所示。

$$H(z)=\frac{Y(z)}{X(z)}=\frac{b_0+b_1z^{-1}+\cdots+b_Nz^{-M}}{a_0+a_1z^{-1}+\cdots+a_Nz^{-N}} \tag{6.4.1}$$

其中b_n,a_n是滤波器系数。假设$a_0=1$，如果$N\geqslant M$，$a_N\neq 0$，这时 IIR 滤波器阶数为 N。IIR 滤波器的差分方程为(6.4.2)式。

$$y(n)=\sum_{i=0}^{M}b_i x(n-i)-\sum_{k=1}^{N}a_k y(n-k) \tag{6.4.2}$$

其特点是信号流图中含有反馈支路，即含有环路。其脉冲响应是无限长的。除了没有线性相位结构外，实现 IIR 滤波器的三种结构与实现 FIR 滤波器相仿，依次为：

（1）直接形式。此形式中，差分方程（6.4.2）式按方程直接实现。这种滤波器可分为两个部分，滑动平均部分和递归部分（或者，分子和分母部分）。根据两部分运算的先后，有两种实现形式，即直接 I 型和直接 II 型结构。

（2）级联形式。分别把等式（6.4.1）中的系统函数 $H(z)$的分子分母因式分解成二阶子系统。每个二阶子系统都以直接形式实现，整个系统函数由分子分母双二阶环节的级联实现。

（3）并联形式。把系统函数 $H(z)$用部分分式展开，合并共轭项，使之成二阶子系统的和。每一子系统用直接形式实现，整个系统函数以子系统的并联网络实现。

本节将简要讨论这些形式。先用系统函数的有理分式（或直接形式结构）描述 IIR 滤波器。然后提供 MATLAB 函数把直接形式结构转换成级联和并联形式结构。

6.4.1 直接型

用延迟元件、倍率器和加法器，以直接形式实现差分方程（6.4.2）的结构称为直接结构。具体说明如下，设 $M=N=2$，那么差分方程为(6.4.3)式。

$$y(n)=b_0x(n)+b_1x(n-1)+b_2x(n-2)-a_1y(n-1)-a_2y(n-2) \tag{6.4.3}$$

其具体实现如图 6.4.1(a)所示，这种信号流图叫做直接 I 型结构。直接 I 型结构先实现有理函数 $H(z)$的分子部分，后实现其分母部分，然后把它们级联起来。分子部分是抽头延迟线，它后面是分母部分，为反馈抽头延迟线。此结构中存在两部分独立的延迟线，因此，需要四个延迟元件。如果交换两部分的连接次序，如图 6.4.1(b)所示，先处理分母部分，再处理分子部分。此时，两个延迟线并排，因此可拿掉其中的一个延迟线。这种缩减构成了另一种标准的结构，叫做直接 II 型结构，它只要两个延迟元件，如图 6.4.1(c)所示，注意，从输入输出的观点看，这两种直接形式是等价的，但在内部，它们的信号是不同的。

直接形式结构由两个系数行向量 b 和 a 描述。信号通过这种滤波器的输出由 filter 函数进行计算，y=filer(b, a, x)。

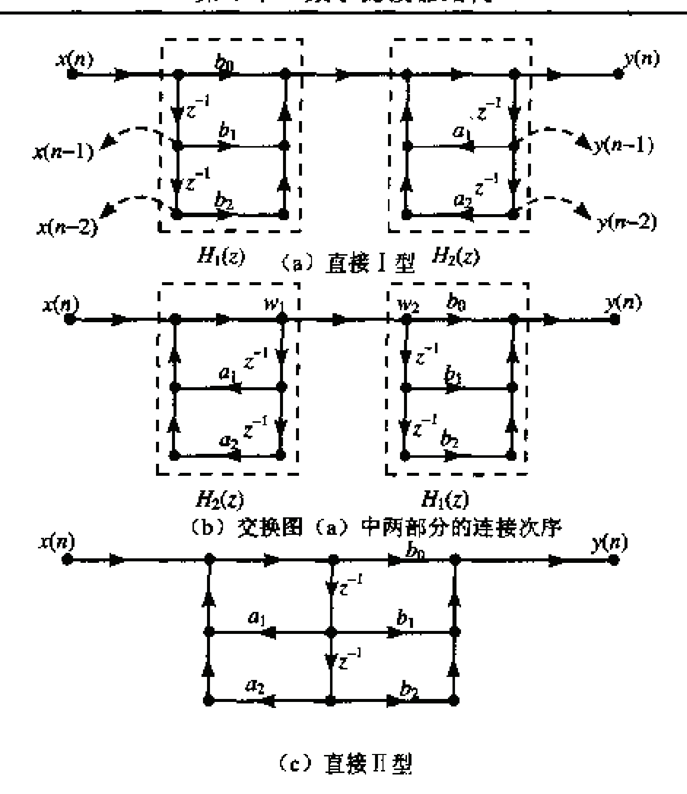


图 6.4.1　二阶 IIR 滤波器直接型结构

6.4.2　级联形式

在这种形式中，系统函数 $H(z)$写成实系数二阶子系统的乘积形式。首先把分子、分母多项式的根解出，然后把每一对共轭复根或任意两个实根组合在一起，得到二阶子系统。在本章下面各节中，假设 N 为偶数，则：

$$H(z)=b_0\frac{1+\dfrac{b_1}{b_0}z^{-1}+\cdots+\dfrac{b_N}{b_0}z^{-N}}{1+a_1z^{-1}+\cdots+a_Nz^{-N}}=b_0\prod_{k=1}^{K}\frac{1+B_{k,1}z^{-1}+B_{k,2}z^{-2}}{1+A_{k,1}z^{-1}+A_{k,2}z^{-2}} \tag{6.4.4}$$

其中 K 等于 $N/2$，$B_{k,1}$，$B_{k,2}$，$A_{k,1}$，$A_{k,2}$为实数，它们表示二阶子系统的系数。二阶子系统为：

$$H_k(z)=\frac{Y_{k+1}(z)}{Y_k(z)}=\frac{1+B_{k,1}z^{-1}+B_{k,2}z^{-2}}{1+A_{k,1}z^{-1}+A_{k,2}z^{-2}};\qquad k=1,\ 2,\ \cdots,K \tag{6.4.5}$$

$H_k(z)$称做第 k 个双二阶环节(biquad)。它的输入是第$(k-1)$个双二阶环节的输出，同时第 k 个双二阶环节的输出为第$(k+1)$个双二阶环节的输入。每一个双二阶环节 $H_k(z)$可用图 6.4.1(c)所示的直接Ⅱ型结构实现。整个滤波器由双二阶环节的级联形式实现：

$$Y_1(z)=b_0X(z);\quad Y_2(z)=H_1(z)\cdot Y_1(z);\quad \cdots;\quad Y(z)=H_K(z)\cdot Y_K(z);$$

因而整个级联系统的传递函数为各个传递函数的连乘

$$Y(z)=H_K(z)\times\cdots\times H_1(z)\cdot Y_1(z)=H(z)Y_1(z)$$

即

$$H(z)=H_K(z)\times\cdots\times H_1(z) \tag{6.4.6}$$

假设 $N=4$，则图 6.4.2 给出此四阶 IIR 滤波器的级联形式结构。

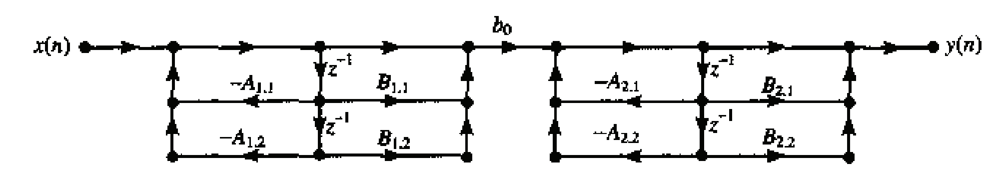


图 6.4.2　N=4 时的级联型结构

如果已知直接形式滤波器的系数 b 和 a，要得到级联形式结构的系数 b_0、$B_{k,i}$和 $A_{k,i}$，可以使用级联型 FIR 滤波器的同－MATLAB 函数 tf2sos.m。它的调用格式为

[sos,G]=tf2sos(b,a)

其中各输入输出变元，特别是 sos 的含义见(6.3.7)式。

简单地讨论一下编写这个子程序的思路。首先通过给较短的向量补零，使向量 b 和 a 一样长。这样做的结果是把负幂系数向量化成了正幂系数向量，从而可以用求根函数 roots。接着，用 z = roots(b)和 p = roots(a) 对向量 b 和 a 分别求根。然后用 cplxpair 函数把这些根以共轭复根对的次序排列，最后用 poly 函数计算每一对共轭复根对应的多项式 $B(z)$和 $A(z)$的值。再把每一对根转换成二阶分子或分母多项式。

信号通过级联形式滤波器后的输出由 sosfilt 函数实现。其调用格式为

y = sosfilt(sos, x)

它的编程思路很简单。把每个二阶环节的系数存在矩阵 B 和 A 中，依次在各个循环里使用，在此循环中采用 filter 函数。“输入”先乘以 b_0，以后每个滤波器的输出作为下一级滤波器的输入，最后一个滤波器的输出即为总的输出。它的核心语句如下：

```
[K,L] = size(B);                          % 分解出 sos 的行数，即级数 K
N = length(x); yk = zeros(K+1,N);         % 预置中间变量yk 矩阵，输入输出共 K=1 行
yk(1,:) = x;                              % 置yk(1,: ) 为输入变量
for i = 1:1:K                             % K 级滤波用 K 次循环实现
      yk(i+1,:) = filter(B(i,:),A(i,:),yk(i,:));    % 第 k 级滤波的计算语句
end
y = b0*yk(K+1,:);                         % yk 中的第 K+1 行就是 y
```

与级联结构有关的函数还有函数 sos2tf。它把级联形式转换成直接形式，是 tf2sos 的逆运算。这是一个包含几次多项式乘法的简单运算。为达到这个目的，只要在 K 次的循环中使用 conv 函数即可，核心语句为

```
b=1;a=1;
for i=1:1:K
        b=conv(b,B(i,:));
        a=conv(a,A(i,:));
end
```

由于 Mathworks 公司开发的软件是商品，要考虑能适应各种情况，测试的要求特别严格，所以程序比较复杂。读者直接打开这些程序，往往和现在本书给出的很不相同，因为本书给出的程序主要考虑的是易于理解概念，并不考虑其运行的效率。

例 6.4.1：一个滤波器由下面的差分方程描述，求出它的级联形式结构。

$16y(\mathrm{n})+12\,y(n-1)+2\,(n-2)\;-4\,y(n-3)\;-y(n-4)=x(n)\;-3x(n-1)+11x(n-2)\;-27x(n-3)+18y(n-4)$

解：MATLAB 程序 hc641 语句如下。

```
b = [1,-3,11, -27,18];
a = [16,12,2, -4, -1];
[sos,G] = tf2sos(b,a)
```

程序运行结果为

```
sos =    1.0000  -3.0000   2.0000   1.0000  --0.2500  --0.1250
         1.0000   0.0000   9.0000   1.0000   1.0000    0.5000
G =      0.0625
```

由此可知滤波器的级联结构方程为

$$H(z)=0.0625\cdot\frac{1-3z^{-1}+2z^{-2}}{1-0.25z^{-1}-0.125z^{-2}}\cdot\frac{1+9z^{-2}}{1+z^{-1}+0.5z^{-2}} \tag{6.4.7}$$

所得的结构如图 6.4.3 所示。

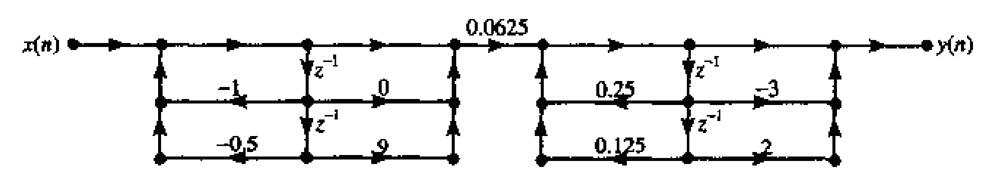


图 6.4.3 例 6.4.1 的级联结构

为了检查级联结构是否正确，用两种形式计算脉冲响应的前八个样本。

```
format long; delta = impseq(0,0,7);    % 输入长度为八的单位脉冲序列
hcas = G*sosfilt(sos,delta)            % 用级联结构的滤波函数 sosfilt 求输出
hdir = filter(b,a,delta)
```

运行结果表明，hdir 的计算结果与 hcas 相同。

6.4.3 并联形式

在这种形式中，系统函数 $H(z)$用部分分式展开为二阶子系统的和的形式。

$$\begin{aligned}H(z)&=\frac{b_0+b_1z^{-1}+\cdots+b_Nz^{-N}}{1+a_1z^{-1}+\cdots+a_Nz^{-N}}=\frac{\bar{b}_0+\bar{b}_1z^{-1}+\cdots+\bar{b}_Nz^{-N}}{1+a_1z^{-1}+\cdots+a_Nz^{-N}}+\sum_{k=0}^{M-N}C_kz^{-k}\\&=\sum_{k=1}^{K}\frac{B_{k,0}+B_{k,1}z^{-1}}{1+A_{k,1}z^{-1}+A_{k,2}z^{-2}}+\sum_{k=0}^{M-N}C_kz^{-k}\end{aligned} \tag{6.4.8}$$

其中：K 等于 $N/2$；$B_{k,0}$；$B_{k,1}$；$A_{k,1}$；$A_{k,2}$ 为实数，它们是二阶子系统的系数；等式右端的第二项称为直接项，它仅在 $M\geqslant N$ 时存在。K 个二阶子系统的通式为

$$H_k(z)=\frac{B_{k,0}+B_{k,1}z^{-1}}{1+A_{k,1}z^{-1}+A_{k,2}z^{-2}};\qquad k=1,2,\cdots,K \tag{6.4.9}$$

$H_k(z)$叫做第 k 阶有理双二阶环节。而它们的输出 $Y_k(z)$之和就是滤波器总的输出 $Y(z)$，如(6.4.10)式所示。

$$Y_k(z)=H_k(z)X(z);\quad Y(z)=\sum_{k=1}^{K}Y_k(z)\qquad M<N \tag{6.4.10}$$

滤波器输入 $X(z)$对所有的双二阶环节均有效，同时若 $M\geqslant N$，$X(z)$也是直接项部分的输入。这些环节的和构成滤波器的输出。因此可用一个并联结构来实现 $H(z)$。每一个双二阶环节 $H_k(z)$可用直接 II 型实现。例如，使 $M=N=4$，此时存在直接项，其值为常数 C_0。图 6.4.4 给出了此四阶 IIR 滤波器的并联形式的结构。

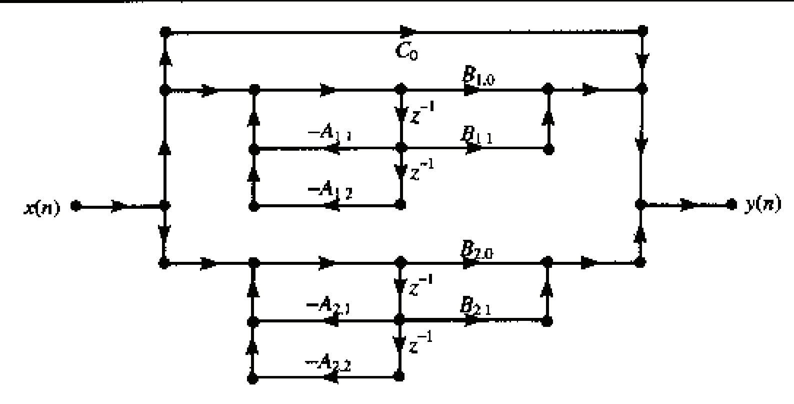


图 6.4.4　N=M=4 时的并联结构

MATLAB 信号处理工具箱没有给出把直接型系数 b 和 a 转换成并联结构系数 A, B 和 C 的专用子程序。它给出的相似的函数是[r,p,C]=residuez(b,a)，因为求留数就等价于把多项式分式化为部分分式的和，也就是并联结构。问题在于 residuez 求出的是单根 p 和对应于单根的留数 r，它们有可能是成对的共轭复数。需要的并联结构系数 A, B 则是二阶环节系数向量，它们必须取实数才能实用。为此必须要把共轭复根 $p_{k,1}$ 和 $p_{k,2}$ 及与它们对应的共轭留数 $r_{k,1}$ 和 $r_{k,2}$ 合成实系数的二阶子系统，如(6.4.11)式。

$$H_k(z)=\frac{r_{k,1}}{1-p_{k,1}z^{-1}}+\frac{r_{k,2}}{1-p_{k,2}z^{-1}}=\frac{B_{k,0}+B_{k,0}z^{-1}}{1+A_{k,1}z^{-1}+A_{k,2}z^{-1}} \tag{6.4.11}$$

实际上，这就是知道一对共轭的极点留数求其多项式系数的问题，它是多项式分解为极点留数的逆运算。这个逆运算和正运算用的是同一个函数 residuez，变换的方向是由输入变元的数目来决定。把多项式分解为极点留数，输入变元是两个，而输出变元是三个；由极点留数求多项式系数，则输入变元是三个，输出变元是两个。所以用 MATLAB 求(6.4.11)式的分子分母系数时只要一条语句：[Bk,Ak]=residuez(rk,pk,Ck)。

其中：*rk*, *pk*, *Ck* 分别是这对共轭留数 *rk*、共轭极点 *pk* 和直接项 *Ck*；*Bk*, *Ak* 分别是(6.4.11)式右端分子和分母多项式的系数。要注意的是此时实际上 *Ck* 不存在，但不能省略它，必须输入一个空矩阵，这才能使 residuez 搞清变换的方向。

参考文献[1]给出了由系统直接结构转为并联型结构的函数 tf2par。也作为本书的子程序其调用格式为[C, B, A] = tf2par(b, a)。

其中所有的变量 C, B, A, b, a 都和(6.4.8)式中的符号相对应。信号通过并联形式的滤波器后的输出可以用 parfilt 函数来计算。tf2par 的反向变换函数为 par2tf,，调用格式为[b,a]= par2tf(C,B,A)。因为信号处理工具箱中没有并联结构的变换函数，下面把这些子程序的主要部分列出。

```
function [C,B,A] = tf2par(b,a);
% 直接型到并联型的转换
M = length(b); N = length(a);
[r1,p1,C] = residuez(b,a);    % 先求系统的单根p1,对应的留数r1 及直接项C
p = cplxpair(p1,1e-9);        % 用配对函数 cplxpair 由p1 找共轭复根p, 1e-9 为容差
I = cplxcomp(p1,p);           % 找p1 变为p 时的排序变化, MATLAB 无此子程序, 列于后面
r = r1(I);                    % 让r1 的排序同样变化成为r,以保持与极点对应
% 变为二阶子系统
K = floor(N/2); B = zeros(K,2); A = zeros(K,3); % 两阶子系统变量的初始化
if K*2 == N;                  % N 为偶, A(z)的次数为奇,有一个因式是一阶的
    for i=1:2:N-2
        pi = p(i:i+1,:);      % 取出一对极点
```

```
            ri = r(i:i+1,:);                       % 取出一对对应的留数
            [Bi,Ai] = residuez(ri,pi,[ ]);         % 二个极点留数转为二阶子系统分子分母系数
            B(fix((i+1)/2),:) = real(Bi);          % 取 Bi 的实部，放入系数矩阵 B 的相应行中
            A(fix((i+1)/2),:) = real(Ai);          % 取 Ai 的实部，放入系数矩阵 A 的相应行中
        end
        [Bi,Ai] = residuez(r(N-1),p(N-1),[ ]);% 处理实单根
        B(K,:) = [real(Bi) 0]; A(K,:) = [real(Ai) 0];
    else                                           %  N 为奇，A(z)的次数为偶,所有因式都是二阶的
      for i=1:2:N-1
            pi = p(i:i+1,:);                       % 取出一对极点
            ri = r(i:i+1,:);                       % 取出一对对应的留数
            [Bi,Ai] = residuez(ri,pi,[ ]);         % 二个极点留数转为二阶子系统分子分母系数
            B(fix((i+1)/2),:) = real(Bi);          % 取 Bi 的实部，放入系数矩阵 B 的相应行中
            A(fix((i+1)/2),:) = real(Ai);          % 取 Ai 的实部，放入系数矩阵 A 的相应行中
        end
    end
```

子程序 tf2par 函数首先用 residuez 函数分解出所有的极点和留数，再用 cplxpair 函数把极点留数按复数共轭极点、实数极点的顺序进行排列。但由于调用此函数改变极点排列后，不能保证极点和它原来的留数对应。因此开发了一个新的 cplxcomp 函数，它找出 $p2$ 的新排序 I，用它重新给留数数组 r 排序。

```
function I = cplxcomp(p1,p2)
% I = cplxcomp(p1,p2)
% 比较两个包含同样标量元素但(可能)有不同下标的复数对
% 本语句必须用在p2 = cplxpair(p1)语句之后，以重新排序其相应的留数向量
%
I=[];
for j=1:length(p2)
   for i=1: length(p1)
     if (abs(p1(i)-p2(j)) < 0.0001) I=[I,i]; end
   end
end, I=I';
```

得到按共轭条件排列的极点-留数对后，tf2par 函数用 residuez 函数的逆形式计算双二阶环节的分子和分母，这时它们一定是实数，然后把它们放入系数矩阵 B 和 A 中。

信号通过并联形式的滤波器后的输出可以用 parfilt 函数来计算。信号处理工具箱中没有这个函数，所以采用参考文献[1]中编写的子程序如下。首先输入经过直通部分 C 的滤波，其结果存在矩阵 yk 的第一行中，随后，计算对相同输入下的所有二阶滤波器的输出，这是通过 for 循环语句分别把每个二阶环节对输入信号进行滤波，并把结果存在矩阵 yk 的各行中，滤波是用 filter 函数进行的。最后，把 yk 中所有各列相加，产生最终的输出 y。

```
function y = parfilt(C,B,A,x);
% 并联型 IIR 滤波器对信号的滤波计算
%
[K,L] = size(B); N = length(x);
yk = zeros(K+1,N);                         % 对各子系统输出矩阵初始化
yk(1,:) = filter(C,1,x);                   % 直接部分的输出
for i = 1:K
 yk(i+1,:) = filter(B(i,:),A(i,:),x);      % 各二阶子系统的输出
end
y = sum(yk);                               % 求总输出
```

MATLAB 信号处理工具箱中没有实系数并联结构的函数。要从并联形式得到直接形式函数，需要开发一个函数 par2tf，它是 tf2par 的逆运算。它首先计算每一个双二阶环节的极点和留数，然后把它们组合成系统的总的极点向量和留数向量，再反向调用 residuez 函数，计算分子和分母系数多项式。程序如下：

```
function [b,a] = par2tf(C,B,A);
```

```
% 并联型到直接型的转换
%
[K,L] = size(A); R = [ ]; P = [ ];          % 变量初始化
for i=1:1:K
    [r,p,k]=residuez(B(i,:),A(i,:)); % 把每个双二阶环节变为单极点留数
    R = [R;r]; P = [P;p];           % 并入系统总的极点向量和留数向量中
end
[b,a] = residuez(R,P,C);            % 用系统总的极点向量和留数向量求分子分母系数
b = b(:)'; a = a(:)';
```

例 6.4.2：考虑例 6.4.1 中的滤波器。

$16y(n)+12y(n-1)+2y(n-2)-4y(n-3)-y(n-4)=x(n)-3x(n-1)+11x(n-2)-27x(n-3)+18y(n-4)$

求出它的并联形式。

解：MATLAB 程序 hc642 如下所示。

```
b = [1, -3,11, -27,18]; a = [16,12,2, -4, -1]; % 原始数据
[C,B,A] = tf2par(b,a)                          % 求并联结构的参数
format long; delta = impseq(0,0,7);
hpar = parfilt(C,B,A,delta)                    % 求并联结构的脉冲响应
hdir = filter(b,a,delta)                       % 求直接形式的脉冲响应
```

程序运行的结果为

```
C = -18
B =  -10.0500  -3.9500
      28.1125  -13.362500
A =   1.0000   1.0000     0.5000
      1.0000  -0.2500    -0.1250
```

为了检查并联结构是否正确，最后两条语句用两种形式计算脉冲响应的前八个样本。它们是完全相同的。最后所得的结构如图 6.4.5 所示。

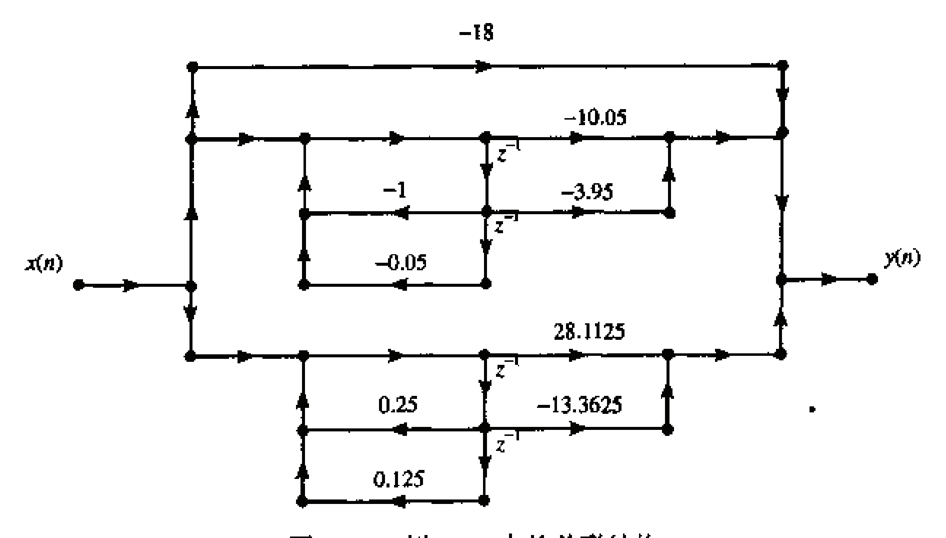


图 6.4.5　例 6.4.2 中的并联结构

在并联型结构中，每一个一阶网络决定一个实数极点，每一个二阶网络决定一对共轭极点，因此调整极点位置方便，但调整零点位置不如级联型方便。另外，各个基本网络是并联的，产生的运算误差互不影响，不像直接型和级联型那样有误差积累。因此，并联形式对运算误差的敏感性最小。由于基本网络并联，如果用硬件实现，可同时对输入信号进行运算，因此并联型结构与直接型和级联型比较，其处理速度最高。

例 6.4.3：在如图 6.4.6 所示的滤波器结构中，包括直接、级联、并联形式的组合。求出这个系统的总特性，分别表示为直接、级联和并联三种结构形式。

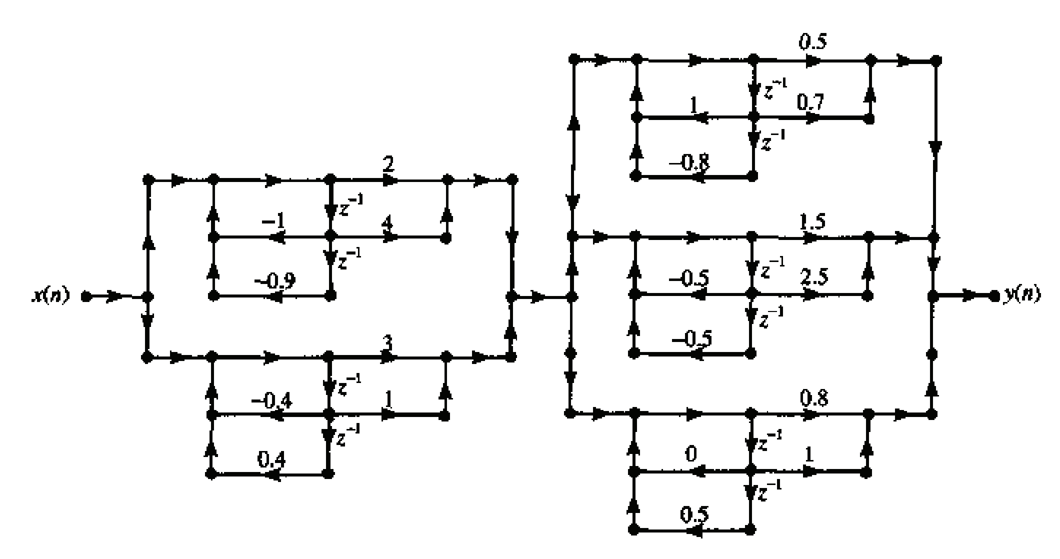


图 6.4.6　例 6.4.3 中的滤波器结构

解：这个结构包含两个并联子系统的级联形式。第一个并联子系统包含两个双二阶环节，而第二个子系统包含三个并联的双二阶环节。用 par2tf 函数把每个并联子系统转换成直接形式，然后把它们级联起来。然后分别对分子、分母多项式进行卷积得到总的直接形式。总的级联和并联形式可由直接形式导出。

```
% 程序 hc643
C = 0; B1 = [2,4;3,1];                       % 两个并联双二阶子系统的分子结构参数
A1 = [1,1,0.9;1,0.4,-0.4];                   % 分母部分
B2 = [0.5,0.7;1.5,2.5;0.8,1];                % 三个并联双二阶子系统的分子结构参数
A2 = [1,-1,0.8;1,0.5,0.5;1,0,-0.5];          % 分母部分
[b1,a1] = par2tf(C,B1,A1);                   % 两个并联子系统合成的直接结构系数
[b2,a2] = par2tf(C,B2,A2);                   % 三个并联子系统合成的直接结构系数
b = conv(b1,b2)                              % 总的直接形式的分子
a = conv(a1,a2)                              % 总的直接形式的分母
[sos,b0] = tf2sos(b,a)                       % 总的级联形式
[C,Bp,Ap] = tf2par(b,a)                      % 总的并联形式
```

程序运行的结果如下：

```
b =
14.000  37.390  27.240  6.2620  12.4810  11.6605  -5.7215  -3.8865  0.5425  0  0
a =
1.000  0.900  0.500  0.080  0.140  0.353  -0.2440  -0.289  -0.182  -0.010  0.072
sos =
   1.0000    1.3762   -0.1854    1.0000    0.2438   -0.3276
   1.0000    0.1025   -0.2747    1.0000    0.1562   -0.6105
   1.0000       0        0       1.0000    0.5000    0.5000
   1.0000   -0.6915    0.6719    1.0000   -1.0000    0.8000
   1.0000    1.8836    1.1328    1.0000    1.0000    0.9000
b0 =    14
C =     0
Bp =
 -20.4201   -1.6000
  24.1602    5.1448
   2.4570    3.3774
  -0.8101   -0.2382
```

```
   8.6129  -4.0439
Ap =
   1.0000   1.0000   0.9000
   1.0000   0.5000   0.5000
   1.0000  -1.0000   0.8000
   1.0000   1.5704   0.6105
   1.0000  -1.1704   0.3276
```

请读者自行写出它们在三种结构形式下的传递函数。这个例子表明，利用 MATLAB 信号处理工具箱和本节开发的函数，可以构造和分析计算很多种滤波器结构。

6.5　一些其他类型的滤波器结构

理论上说，为了实现同一种传递函数，可以有无数种结构形式。选择不同结构形式的依据主要是，应考虑在不同的应用条件下，使参数调整方便和参数稳定。一是希望可调的参数少，用最少的参数设定直接改变设计者所关心的系统指标，而不影响其他指标；二是希望因滤波器参数量化造成的误差对系统指标的影响降低到最小。各种结构的滤波器的计算公式各不相同，不可能在课程中一一介绍。在课程中只能使读者掌握普遍性的方法，重点就是信号流图解析法和下节要介绍的状态空间法。这两种方法与计算机的结合大大增强了它们的功能，可以方便地解决各种滤波器的计算问题。

6.5.1　全零点格型（Lattice）滤波器

格型滤波器广泛应用于数字语音处理和自适应滤波器实践中。在这些应用中，它比其他 FIR 或 IIR 滤波器结构更优越。因为在语音分析和语音合成中，它可用较少数目的系数为各谐振峰实时建模。描述 FIR 滤波器的是全零点格型结构，而描述 IIR 滤波器的是格型梯形结构。

长度为 M（或阶数为 M-1）的 FIR 滤波器具有 M-1 级格型结构，如图 6.5.1 所示。每一级的输入输出通过顺序递推方程建立联系：

$$\begin{aligned} f_m(n) &= f_{m-1}(n) + K_m g_{m-1}(n-1) \\ g_m(n) &= K_m f_{m-1}(n) + g_{m-1}(n-1) \end{aligned} \qquad m=1, 2, \cdots, M-1 \tag{6.5.1}$$

其中：参数 Km，m=1 ,2 ,. $\cdots$, M-1，叫做反射系数，它们是格型滤波器的系数。如果初值 $f0(n)$和 $g0(n)$均为滤波器输入 $x(n)$的倍数（乘以 $K0$），则 M-1 级格型滤波器的输出与（M-1）阶 FIR 滤波器输出一致，也就是：

$$\begin{aligned} f_0(n) &= g_0(n) = K_0 x_0(n) \\ y(n) &= f_{M-1}(n) \end{aligned} \tag{6.5.2}$$

(6.5.1)和 (6.5.2)式的结构可以用图 6.5.1 表示，称为全零格型结构滤波器。它有 M-1 个参数 Km，(m=1, 2, $\cdots$, M-1)，称为反射系数。

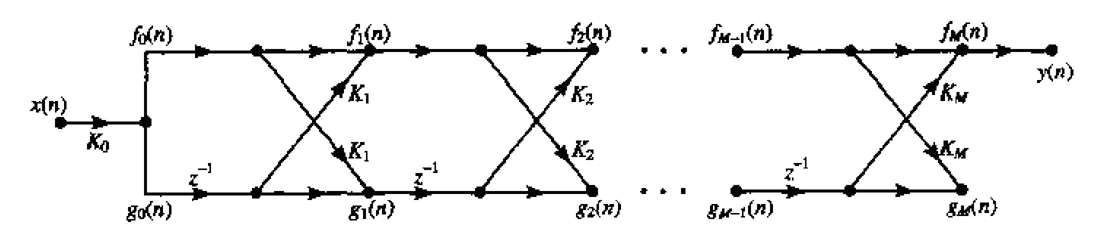


图 6.5.1　全零格型滤波器

如果 FIR 滤波器以直接形式给出：

$$H(z)=\sum_{m=0}^{M-1}b_n z^{-n}=b_0\left(1+\sum_{m=1}^{M-1}\frac{b_m}{b_0}z^{-m}\right) \tag{6.5.3}$$

设多项式为

$$A_{M-1}(z)=1+\sum_{m=1}^{M-1}\frac{b_m}{b_0}z^{-m}=1+\sum_{m=1}^{M-1}\alpha_{M-1}(m)z^{-m} \tag{6.5.4}$$

那么，格型滤波器系数 K_m 可由下面的递推算法得到：

$$\begin{aligned}
&K_0=b_0\\
&K_{M-1}=\alpha_{M-1}(M-1)\\
&J_m(z)=z^{-m}A_m(z^{-1}) \qquad\qquad m=M-1,\cdots,2,1\\
&A_{m-1}(z)=\frac{A_m(z)-K_mJ_m(z)}{1-K_m^2}\\
&K_m=\alpha_m(m)
\end{aligned} \tag{6.5.5}$$

如果在递推中出现有任一个$|K_m|=1$ $(m=1,2,\cdots,M-1)$，则上述算法不成立。由于线性相位滤波器的对称性，$b_0=|b_{M-1}|$由此导出

$$|K_{M-1}|=|\alpha_{M-1}(M-1)|=|b_{M-1}/b_0|=1 \tag{6.5.6}$$

可见，线性相位滤波器不能用格型滤波器实现。

递推公式(6.5.5)相当烦琐，只有专门研究格型滤波器的人才有必要去掌握它。以这个递推公式为基础，MATLAB 信号处理工具箱提供了 tf2latc 函数，只要知道 FIR 滤波器的直接形式系数向量 b，就可以调用这个函数求出格型滤波器的反射系数向量 K，但要注意，向量 b 的首项必须归一化为 1，故应输入

K=t f 2latc (b/b(1))

信号处理工具箱也提供了它的逆函数，已知格型滤波器的反射系数向量 K 时，可以求出它的直接形式系数向量 b。其语句为

b=latc2tf(K)

信号处理工具箱也提供了信号 x 通过格型滤波器滤波后输出 y 的计算函数 latcfilt。

y=latcfilt(K,x)

例 6.5.1：FIR 滤波器由差分方程 $y(n)=x(n)+\frac{13}{24}x(n-1)+\frac{5}{8}x(n-2)+\frac{1}{3}x(n-3)$ 给定，求出它的格型形式。

解：MATLAB 程序 hc651

```
b=[1,13/24,5/8,1/3];                % 原始数据，b 必须归一化
K=tf2latc(b)                        % 计算格型滤波器反射系数
delta = impseq(0,0,3);              % 给出输入脉冲序列
hdirect=filter(b,1,delta)           % 用直接型计算输出响应
hlattice=latcfilt(K,delta)          % 用格型参数计算输出响应
```

运行结果为

$K=$　　0.2500　　0.5000　　0.3333

即 $K_1=1/4$，$K_2=1/2$，$K_3=1/3$。

本例的直接形式和格型形式如图 6.5.2 所示。为了检查格型结构是否正确，上述程序的最后两条语句是用两种形式计算滤波器的脉冲响应，其结果是相同的，说明变换是等价的。

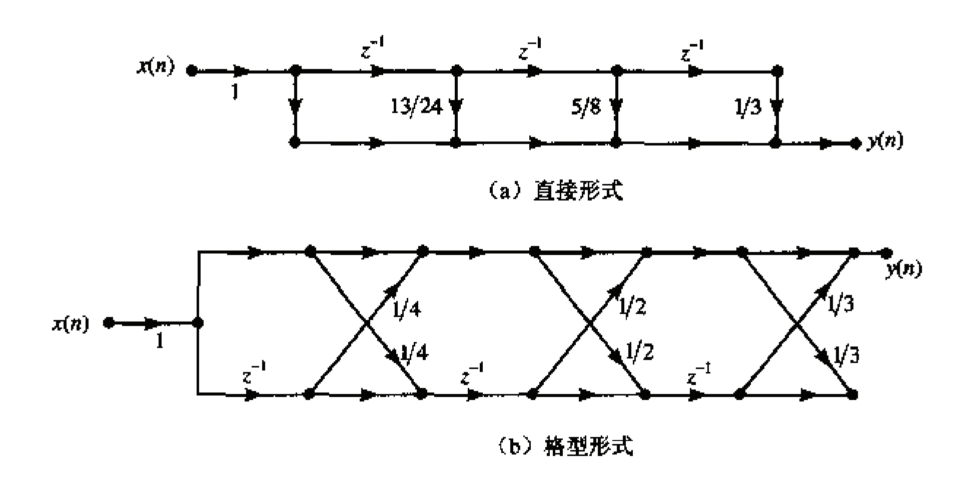


图 6.5.2　例 6.5.1 中的 FIR 滤波器结构

6.5.2　全极点格型滤波器

IIR 滤波器的格型结构只适用于全极点的系统函数，可根据 FIR 格型结构来开发它。设一个全极点系统函数由式 6.5.7 给定。

$$H(z)=\frac{1}{1+\sum_{m=1}^{N}a_N z^{-m}} \tag{6.5.7}$$

与（6.5.4）式相比，可知 $H(z)=1/A_N(z)$ 。显然，它是图 6.5.1 所示的 FIR 格型结构的逆系统。这个 N 阶 IIR 滤波器具有如图 6.5.3 所示的 N 级格型结构。

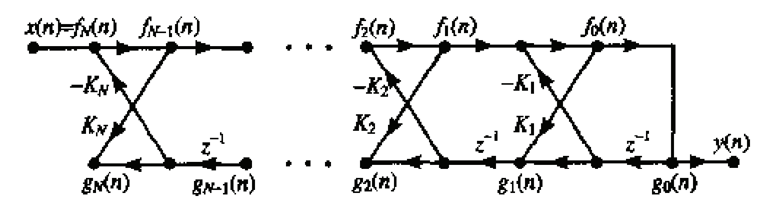


图 6.5.3　全极点格型结构滤波器

每一级的输入输出之间有如下的递推关系：

$$\begin{aligned}
&f_N(n)=x(n)\\
&f_{m-1}(n)=f_m(n)-K_m g_{m-1}(n-1)\\
&g_m(n)=K_m f_{m-1}(n)+g_{m-1}(n-1)\\
&y(n)=f_0(n)=g_0(n)
\end{aligned}\qquad m=N-1,\ldots,2,1 \tag{6.5.8}$$

其中参数 K_m，m=1 ,2 ,...,N-1，是全极点格型的反射系数，各系数可通过（6.5.5）式递推得到。

既然，与 FIR 格型滤波器相同，IIR 滤波器系数同样可由（6.5.5）导出。因此仍可使用信号处理工具箱中的 tf2latc 函数。必须记住数组 $K=[K_1,K_2,\cdots,K_{N-1}]$，而 K_0=1 不包括在 K 内也可使用 latc2tf 函数把格型 K_m 系数转换成直接形式 $a_N(m)$。

例 6.5.2：全极点 IIR 滤波器由下式给出，求出它的格型结构。

$$H(z)=\frac{1}{1+\frac{13}{24}z^{-1}+\frac{5}{8}z^{-2}+\frac{1}{3}z^{-3}}$$

解： MATLAB 程序为：

```
a=[1,13/24,5/8,1/3];
K=tf2latc(a)
```

运行结果：

```
K = 0.2500    0.5000    0.3333
```

也就是说 $K_1 = 1/4$，$K_2 = 1/2$，$K_3 = 1/3$。IIR 滤波器的格型形式和直接形式如图 6.5.4 所示。

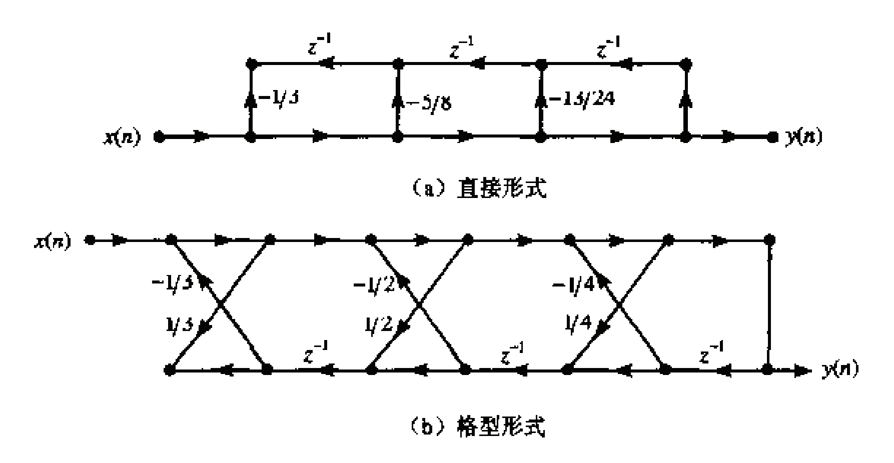


图 6.5.4　例 6.5.2 中的 IIR 滤波器结构

一般的 IIR 滤波器既包含零点，又包括极点，它可用全极点格型作为基本框架来实现。假定 IIR 滤波器的系统函数为

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{6.5.9}$$

假设 $N \geqslant M$，为构造一个格型结构，首先根据（6.5.9）式的分母，以系数 K_m，$1 \leqslant m \leqslant N$ 实现全极点格型，如图 6.5.3 所示然后，增加一个梯形部分，把输出看做 $g_m(n)$ 的加权线性组合。图 6.5.5 所示是 $M=N$ 时的情况。它是具有格型梯形结构的零点-极点 IIR 滤波器。其输出为：

$$y(n) = \sum_{m=0}^{M} C_m g_m(n) \tag{6.5.10}$$

其中 C_m 确定系统函数 $H(z)$的分子部分，称为梯形系数。

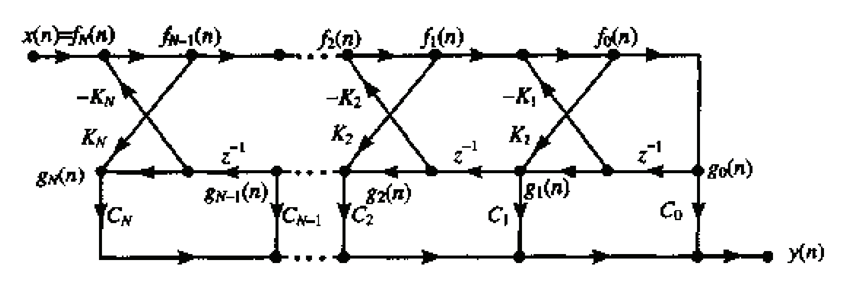


图 6.5.5　兼有零极点的 IIR 滤波器的格型梯形结构

IIR 滤波器格型梯形结构参数与传递函数系数向量之间的关系仍可由 tf2latc 函数求得。其调用格式为 [K, C]=tf2latc(b, a)。

例 6.5.3：把用下式表示的含零极点的 IIR 滤波器转换为格型梯形结构。

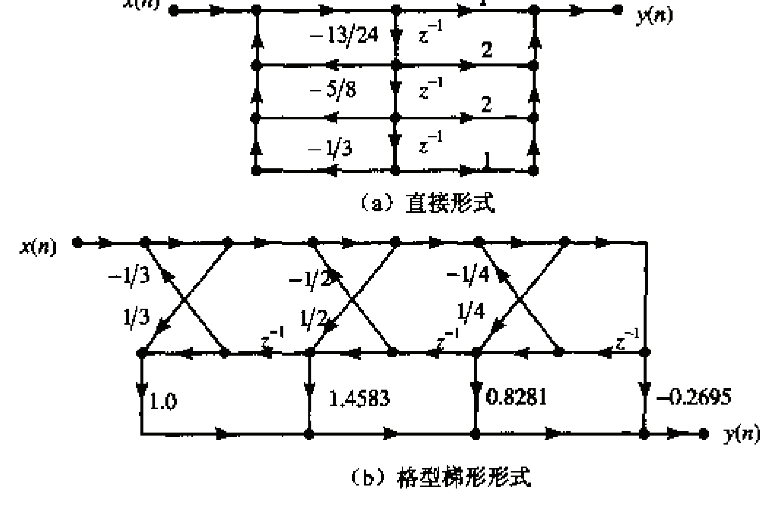
$$H(z)=\frac{1+2z^{-1}+2z^{-2}+z^{-3}}{1+\frac{13}{24}z^{-1}+\frac{5}{8}z^{-2}+\frac{1}{3}z^{-3}}$$

解：MATLAB 程序为

```
b=[1,2,2,1];a=[1,13/24,5/8,1/3];
[K,C]=tf2latc(b,a)
```

运行结果为：

```
K = 0.2500        C = - 0.2695
    0.5000            0.8281
    0.3333            1.4583
                      1.0000
```



(a) 直接形式

(b) 格型梯形形式

图 6.5.6　例题 6.5.3 中的结构

所得的直接形式和格型梯形形式结构如图 6.5.6 所示。为检查格型梯形结构是否正确，用两种形式计算它的脉冲响应的前八个样本。格型梯形滤波器的输入输出关系仍然由 latcfilt 函数计算。程序如下：

```
[x,n]=impseq(0,0,7);format long
hdirect=filter(b,a,x)
hlatc=latcfilt(K,C,x)
```

运行结果为说明，hdirect 与 hlatc 是相同的

6.5.3　用信号流图与 MATLAB 结合求解格型滤波器

现在用格型梯形滤波器为典型，再试用一次第 6.2.2 节介绍的信号流图法。

例 6.5.4：求图 6.5.7 所示的格型梯形滤波器的传递函数。

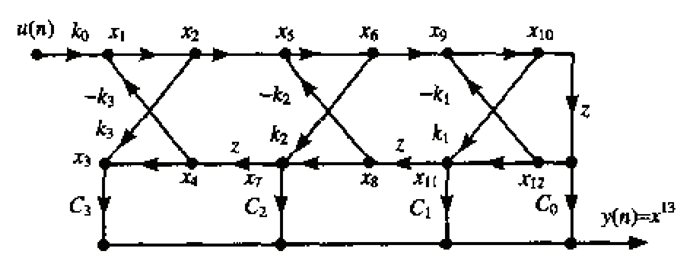


图 6.5.7　二阶全极点格型滤波器的结构图

解：从图 6.5.7 可以列出信号流图方程：

$x_1=u-k_3x_4$;　$x_2=x_1$;　$x_3=k_3x_2+x_4$; $x_4=qx_7$; $x_5=x_2-k_2x_8$;　$x_6=x_5$;

$x_7=k_2x_6+x_8$;　$x_8=qx_{11}$; $x_9=x_6-k_1x_{12}$; $x_{10}=x_9$;　$x_{11}=k_1x_{10}+x_{12}$; $x_4=qx_{10}$;

$x_{13}=y=C_0x_{12}+C_1x_{11}+C_2x_7+C_3x_3$

例中的参数已给定数值如下：

$k_0=1, k_1=1/4, k_2=1/2, k_3=1/3, C_0=-0.2, C_1=0.8, C_2=1.5, C_3=1$。

为了节省篇幅，不必列出 Q 和 P 的矩阵形式，可以按其下标规律直接进行元素赋值。

本例虽然是数值计算，但计算的内容中带有 z 变换算子 $q=z^{-1}$,所以 Q 矩阵仍然必须用符号属性，对 Q 赋值时第一个元素必须取含 q 的算式。程序如下：

```
% MATLAB 程序 hc654
clear, syms q                                  % 对符号变量进行定义
k1=1/4; k2=1/2; k3=1/3; k0=1;                  % 给网络中的常数
C0=-0.2; C1=0.8; C2=1.5; C3=1;
Q(4,7)=q;                                      % Q 的第一个赋值元素为符号变量
Q(5,2)=1; Q(1,4)=-k3; Q(2,1)=1;                % 为连接矩阵的非零元素赋值
Q(3,2)=k3; Q(3,4)=1; Q(5,8)=-k2;
Q(6,5)=1; Q(7,6)=k2; Q(7,8)=1;
Q(8,11)=q; Q(9,6)=1;
Q(9,12)=-k1; Q(10,9)=1;
Q(11,10)=k1; Q(11,12)=1; Q(12,10)=q;
Q(13,3)=C3; Q(13,7)=C2;
Q(13,11)=C1; Q(13,12)=C0;
Q(13,13)=0; P(1,1)=k0; P(13,1)=0;              %给右下角元素赋值，使未赋值元素全部为零
W=inv(eye(size(Q))-Q)*P                        % 信号流图方程解
pretty(W(13))                                  % 以美观的形式显示 W 中第 13 行
```

程序运行后的结果为 $W(13)=\dfrac{24q^3+49q^2+42.9q+30.8}{8q^3+15q^2+13q+24}$。将 $q=z^{-1}$ 代入，按 z 的降幂排列，可得：

$$H(z)=\frac{Y(z)}{U(z)}=\frac{\dfrac{30.8}{24}+\dfrac{42.9}{24z^{-1}}+\dfrac{49}{24z^{-2}}+z^{-3}}{1+\dfrac{13}{24z^{-1}}+\dfrac{5}{8z^{-2}}+\dfrac{1}{3z^{-3}}}$$

6.6 状态变量分析法

在研究系统内部结构以及内部变量随输入信号变化的特性时，往往需要引入状态变量分析法，它可以使用矩阵、线性代数等数学工具进行分析研究。MATLAB 中用得很多的 filter 函数就是利用状态变量方程编写的。MATLAB 的信号处理工具箱也提供了状态变量与其他各种网络结构之间的变换函数，还有一些网络结构只能用状态变量分析法进行描述。其实，输入输出分析法、信号流图法和状态变量法都是线性系统的基本分析方法，它们分别从系统内外的不同的侧面进行分析研究，相互补充。

6.6.1 状态方程和输出方程

状态变量分析法有两个基本方程，即状态方程和输出方程。状态方程把系统内部的状态变量和输入信号联系起来；而输出方程则把输出信号和状态变量联系起来。

按信号流图，系统内部有很多信号节点，但并不把它们都定为状态变量节点。要确定一组最少的节点变量，使得它们之间是线性无关的，即任何一个状态变量不能由其他状态变量以线性组合的方式构成。在滤波器的三个基本组成元件中，倍率器和相加器都属于线性组合运算，它们的输出不可能形成新的状态变量。延迟器不属于线性组合运算，所以系统的结构图中，状态变量的数目等于简化到最低限度的延迟器数目。选择状态变量的基本原则就是把每个迟延环节输出端的变量作为状态变量。以每个状态变量为主体可以写出一个状态方程，若有 N 个状态变量，就可以列出 N 个状态方程，构成 N 阶状态空间方程组。输出变量既然是

状态变量的线性组合，所以一定能用状态变量的线性方程来表示，称为输出方程。

现在以四阶 IIR 直接型滤波器来说明它的状态方程列写方法。其差分方程为

$$\begin{aligned} y(n) = &-a_1 y(n-1) - a_2 y(n-2) - a_3 y(n-3) - a_4 y(n-4) + \\ &b_0 u(n) + b_1 u(n-1) + b_2 u(n-2) + b_3 u(n-3) + b_4 u(n-4) \end{aligned} \tag{6.6.1}$$

习惯上都把状态变量用 x_i (i=1, 2, …, N)表示，因此在式中，$u(n)$作为输入，$y(n)$是输出。因为直接 I 型的延迟环节并没有化简至最少，系统的结构图用直接 II 型表示为图 6.5.8。其中在四个迟延器后的变量分别为 $x_1(n)$, $x_2(n)$, $x_3(n)$和 $x_4(n)$，现选择它们作为此系统的四个状态变量。而在各个迟延支路前端的变量应该是 $x_1(n+1)$, $x_2(n+1)$, $x_3(n+1)$和 $x_4(n+1)$。它们应该可以用状态变量的线性组合(6.6.2)式来表示。

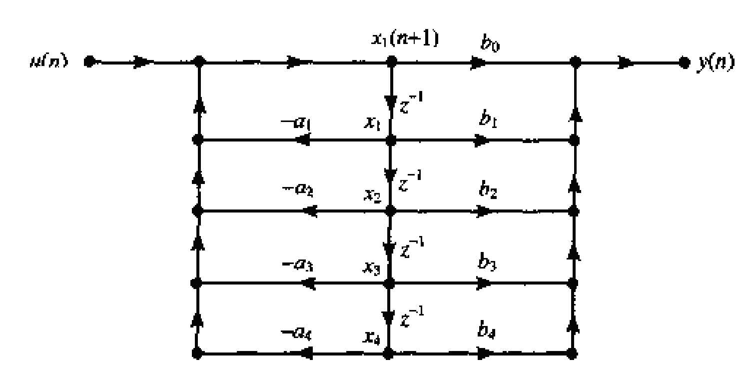


图 6.5.8　四阶 IIR 滤波器的直接 II 型结构

$$\begin{aligned} x_1(n+1) &= u(n) - a_1 x_1(n) - a_2 x_2(n) - a_3 x_3(n) - a_4 x_4(n) \\ x_2(n+1) &= x_1(n) \\ x_3(n+1) &= x_2(n) \\ x_4(n+1) &= x_3(n) \end{aligned} \tag{6.6.2}$$

输出方程也可从图 6.5.8 中得到，把 $x(n+1)$代换掉，整理成状态变量的线性组合，可得

$$\begin{aligned} y(n) &= b_0 x(n+1) + b_1 x_1(n) + b_2 x_2(n) + b_3 x_3(n) + b_4 x_4(n) \\ &= b_0 [u(n) - a_1 x_1(n) - a_2 x_2(n) - a_3 x_3(n) - a_4 x_4(n)] + b_1 x_1(n) + b_2 x_2(n) + b_3 x_3(n) - b_4 x_4(n) \end{aligned}$$

最后得到输出方程(6.6.3)式。

$$y(n) = (b_1 - a_1)x_1(n) + (b_2 - a_2)x_2(n) + (b_3 - a_3)x_3(n) + (b_4 - a_4)x_4(n) + b_0 u(n) \tag{6.6.3}$$

(6.6.2)式和(6.6.3)式就构成了这个系统的状态方程和输出方程。用矩阵表示为

$$X(n+1) = \begin{bmatrix} x_1(n+1) \\ x_2(n+1) \\ x_3(n+1) \\ x_4(n+1) \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & -a_3 & -a_4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \\ x_4(n) \end{bmatrix} + \begin{bmatrix} b_0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot u(n)$$

$$y(n) = \begin{bmatrix} (b_1 - a_1) & (b_2 - a_2) & (b_3 - a_3) & (b_4 - a_4) \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \\ x_4(n) \end{bmatrix} + b_0 u(n)$$

这样就得到了差分状态方程的标准形式：

$$X(n+1) = AX(n) + BU(n) \tag{6.6.4}$$

$$Y(n) = CX(n) + DU(n) \tag{6.6.5}$$

其中：

$$X(n)=\begin{bmatrix}x_1(n)\\x_2(n)\\x_3(n)\\x_4(n)\end{bmatrix},\quad A=\begin{bmatrix}-a_1 & -a_2 & -a_3 & -a_4\\1&0&0&0\\0&1&0&0\\0&0&1&0\end{bmatrix},\quad B=\begin{bmatrix}b_0\\0\\0\\0\end{bmatrix} \tag{6.6.6}$$

$$C=[(b_1-a_1)\ \ (b_2-a_2)\ \ (b_3-a_3)\ \ (b_4-a_4)],\quad D=b_0$$

方程式(6.6.4)和 (6.6.5)是系统的状态方程组。其中(6.6.4)式称为状态方程，(6.6.5)式称为输出方程。A, B, C, D 是系统状态方程的基本系数矩阵。知道这四个系数矩阵，系统的性能就惟一地确定了。如果系统是 N 维的，即有 N 个延迟环节，因而有 N 个状态变量。又有 K 个输出变量，L 个输入变量，则变量 X 是 $N\times1$ 阶向量，Y 是 $K\times1$ 阶向量，U 是 $L\times1$ 阶向量，而系数 A 是 $N\times N$ 阶的，B 是 $N\times L$ 阶的，C 是 $K\times N$ 阶的，D 是 $K\times L$ 阶的。本例中，只有一个输出 u 和一个输入 y，$K=L=1$。所以 A 是 4×4 阶的，B 是 4×1 阶的，C 是 1×4 阶的，D 是 1×1 阶的标量。

6.6.2　状态方程和其他结构参数的转换

由状态方程的四个参数矩阵很容易求出等价的其他形式结构的参数，首先来推导状态空间的输入输出关系式，得出由状态空间变换为传递函数的公式。

利用 z 变换算子表示状态方程组(6.6.4)可以写出 $X(n+1)=zX(n)=AX(n)+BU(n)$，将 $X(n)$移到左端，按矩阵算法合并，得到 $(zI-A)X(n)=BU(n)$，对方程两端做 z 变换，移项得

$$\frac{X(z)}{U(z)}=(zI-A)^{-1}B$$

因为传递函数 $H(z)$是输入输出的 z 变换之比，将输出方程(6.6.5)代入，可以得到：

$$H(z)=\frac{Y(z)}{U(z)}=C\frac{X(z)}{U(z)}+D=C(zI-A)^{-1}B+D \tag{6.6.7}$$

得到 $H(z)$以后，就很容易得到零极增益或极点留数等表示式了。

由传递函数转换为状态空间参数矩阵的逆运算不是惟一的。因为同一个系统，选择的状态变量不同，就会得出不同的状态方程，因而具有不同的系数矩阵。所以还规定了几种规范形式的状态方程，并且还有从一种状态方程形式变换为另一种状态方程的变换关系，这些变换都是很烦琐的。

MATLAB 提供了各种形式的系统参数之间的变换关系，如表 6.6.1 所示。要特别注意这个表概括了连续系统和离散系统的变换，所以输入变元的意义在两种情况下可能不同。例如传递函数的系数向量 a 和 b 在连续系统是用正幂排列，在离散系统则用负幂排列。在具体使用某个函数时，建议读者还是要用 help 命令阅读其使用方法和输入输出变元的定义，以免造成错误。

表 6.6.1　用于线性系统类型和滤波器结构之间转换的工具箱函数表

	传递函数 b,a	状态空间 A，B，C，D	零极增益 z，p，k	部分分式 r，p，h	格型结构 K，V	级联结构 sos	并联结构
传递函数 b,a		tf2 ss	tf2zp roots	residue residuez	tf2l atc	tf2s os	dir2 par*
状态空间	ss2tf	ss2 ss	ss2zp			ss2 sos	
零极增益	zp2tf　poly	zp2ss				zp2sos	
部分分式 r，p，h	Residue residuez						
格型结，K，V	latc2tf						
级联结构 sos	sos2tf	sos2ss	sos2zp				
并联结构	par2dir*						

6.6.3　信号流图方程与状态方程的关系

信号流图方程组和状态空间方程组都可以完整地表示离散线性系统，两者是完全等价的。其差别在于：

（1）信号流图把系统结构中的全部变量都包括在方程中。虽然每个方程很简单，但方程的数目却很多。状态空间方程中的变量是经过挑选的线性无关的那部分，因此变量数和方程数都大大减少。

（2）信号流图的左端是各个变量的当前值，在方程的右端则把延迟算子 $q=z^{-1}$ 当做一般的乘子同等看待，不再区分变量的当前值或移位后的值。状态空间方程的左端是各个状态变量的未来值，而在方程的右端则是各个状态变量的当前值。用的时移算子为提前算子 z，集中在方程的左端。这意味着根据右端的当前状态变量可以计算下一时刻的未来状态变量。所以状态空间方程也反映了一种递推关系，便于编写递推程序。

（3）如果结构图中出现了没有迟延环节的环路（也称为代数环），就破坏了递推的可能，因而会出现列不出状态方程的状况。

（4）用状态方程要用手工做许多变量代换推导工作，比较麻烦，并容易出错；信号流图方程组则是非常直接的，只要按流图列写了方程，在填写连接矩阵时不出错误，其他的计算完全由计算机来完成，既简便又准确。在实用中要根据任务的需要来选择方法。

如果要从信号流图求状态空间方程，只要把信号流图中的全部延迟算子 $q=z^{-1}$ 找到，将延迟算子后的变量取为状态变量 $X(n)$，而把延迟算子前的变量取为状态变量的未来值，放在方程左端，写成 $zX(n+1)$，剩下的问题是在方程组的右端通过变量置换，把其余变量都用状态变量表示即可。

6.6.4　例题：正余弦序列产生器

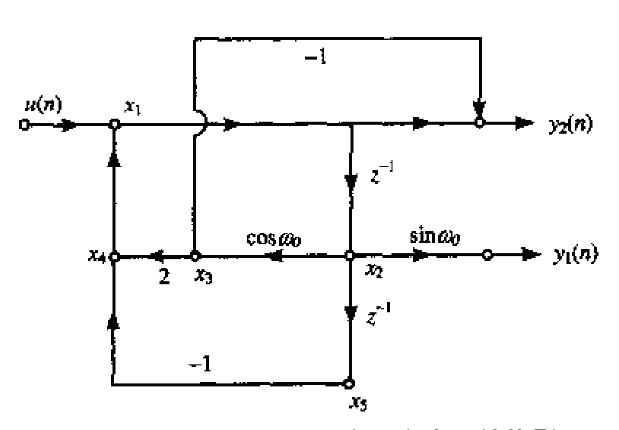


图 6.6.1　数字正余弦序列产生器结构图

滤波器的极点必须配置在单位圆的内部，以保证它们的稳定性和暂态过程的尽快衰减。如果把极点配置到单位圆的圆周上，那么滤波器会产生持续的固定频率的振荡，于是滤波器就变成了振荡器。利用这个特点，可以构成正余弦序列的产生器。

例 6.6.1：图 6.6.1 为一个数字正余弦信号发生器的系统结构。分别用信号流图方程组和状态空间方程组分析此系统的传递函数，证明它在脉冲激励 $u(n)=\delta(n)$下能产生持续的正余弦信号。

解：按信号流图，可以列出系统的方程组：

$$\begin{aligned}
x_1 &= u + x_4 \\
x_2 &= z^{-1}x_1 = q\,x_1 \\
x_3 &= x_2\cos\omega_0 \\
x_4 &= 2x_3 - x_5 \\
x_5 &= z^{-1}x_2 = q\,x_2 \\
x_6 &= y_1 = x_2\sin\omega_0 \\
x_7 &= y_2 = x_1 - x_3
\end{aligned} \tag{6.6.8}$$

写成矩阵形式为

$$X=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\\x_6\\x_7\end{bmatrix}=\begin{bmatrix}0&0&0&1&0&0&0\\q&0&0&0&0&0&0\\0&\cos\omega_0&0&0&0&0&0\\0&0&2&0&-1&0&0\\0&q&0&0&0&0&0\\0&\sin\omega_0&0&0&0&0&0\\1&0&-1&0&0&0&0\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\\x_6\\x_7\end{bmatrix}+\begin{bmatrix}1\\0\\0\\0\\0\\0\\0\end{bmatrix}\cdot u=QX+Pu \tag{6.6.9}$$

按此矩阵，写出 MATLAB 程序 hc661 如下：

```
syms q w0
    Q(2,1) = q; Q(1,4)=1;Q(3,2)=cos(w0);
    Q(4,3)=2; Q(4,5) = -1;
    Q(5,2)=q; Q(6,2) = sin(w0);
    Q(7,1) =1; Q(7,3)= -1; Q(7,7)=0;
    P(1,1)=1; P(7,1)=0;
    W=inv(eye(size(Q)) -Q)*P
```

程序运行的结果为

```
W=   [                1/(1-2*q*cos(w0)+q^2)]
[               q/(1-2*q*cos(w0)+q^2)]
[        q*cos(w0)/(1-2*q*cos(w0)+q^2)]
[ -q*(2*cos(w0)-q)/(-1+2*q*cos(w0)-q^2)]
[             q^2/(1-2*q*cos(w0)+q^2)]
[        q*sin(w0)/(1-2*q*cos(w0)+q^2)]
[    (q*cos(w0)-1)/(-1+2*q*cos(w0)-q^2)]
```

应该关心的是 W 中的第六列和第七列，键入 pretty(W(6)), pretty(W(7)),得到：

$$W(5)=\frac{Y_1(z)}{U(z)}=\frac{q\sin(w_0)}{1-2q\cos(w_0)+q^2}=\frac{z^{-1}\sin(w_0)}{1-2z^{-1}\cos(w_0)+z^{-2}}$$

(6.6.10)

$$W(6)=\frac{Y_2(z)}{U(z)}=\frac{1-q\cos(w_0)}{1-2q\cos(w_0)+q^2}=\frac{1-z^{-1}\cos(w_0)}{1-2z^{-1}\cos(w_0)+z^{-2}}$$

由于 $1-2z^{-1}\cos(\omega_0)+z^{-2}=(1-e^{j\omega_0})(1-e^{-j\omega_0})$

可见系统的极点在单位圆圆周的 $e^{\pm j\omega_0}$ 处，这样的结构在一个初始激励下可以产生持续的正余弦信号。确实，如果加入一个激励 $x(n)=A\,\delta(n)$，它的 z 变换为 A，则得出

$$Y_1(z)=H_1(z)X(z)=\frac{A(\sin\omega_0)z^{-1}}{1-(2\cos\omega_0)z^{-1}+z^{-2}} \tag{6.6.11a}$$

$$Y_2(z)=H_2(z)X(z)=\frac{A[1-(\cos\omega_0)z^{-1}]}{1-(2\cos\omega_0)z^{-1}+z^{-2}} \tag{6.6.11b}$$

它们分别是表 5.1.1 中序列 $a^n\sin\omega_0 n$ 和 $a^n\cos\omega_0 n$ 在 a=1 时的 z 变换。现在由于 $a=1$，所以输出的 y_1 和 y_2 分别为正弦和余弦序列 $\sin\omega_0 n$ 和 $\cos\omega_0 n$。

下面研究如何用状态方程来描述这个系统。先把 $q=z^{-1}$ 的后端变量选择为状态变量，有两个 q，应该有两个状态变量 x_2 和 x_5。再把 $q=z^{-1}$ 的前端变量 x_1 和 x_2 写成状态变量 x_2 和 x_5 的未来值 zx_2 和 zx_5，置于方程左端。即 $x_2=qx_1$ 等价于 $x_1=zx_2$，$x_5=qx_2$ 等价于 $x_2=zx_5$。再把等式右端通过变量置换，消去 x_1，x_3，x_4，x_6，x_7，

只留下状态变量 x_2,x_5 和输入变量 u，于是得到

$zx_2=x_1=u+x_4=2x_3-x_5+u=2x_2\cos\omega_0-x_5+u$

$zx_5=x_2$

$y_1=x_2\sin\omega_0$

$y_2=x_1-x_3=u+x_4-x_2\cos\omega_0=-x_2\cos\omega_0+2x_3-x_5+u=x_2\cos\omega_0-x_5+u$

写成矩阵形式的状态方程为

$$zX=z\begin{bmatrix}x_2\\x_5\end{bmatrix}=\begin{bmatrix}x_2(n+1)\\x_5(n+1)\end{bmatrix}=\begin{bmatrix}2\cos(\omega_0) & -1\\1 & 0\end{bmatrix}\begin{bmatrix}x_2\\x_5\end{bmatrix}+\begin{bmatrix}1\\0\end{bmatrix}u=AX+Bu$$

$$Y=\begin{bmatrix}y_1\\y_2\end{bmatrix}=\begin{bmatrix}\sin(\omega_0) & 0\\\cos(\omega_0) & -1\end{bmatrix}\begin{bmatrix}x_2\\x_5\end{bmatrix}+\begin{bmatrix}0\\1\end{bmatrix}u=CX+Du$$

系统函数应为 $H(z)=C(zI-A)B+D$。

用符号运算方法给 A,B,C,D 赋值，再按上式求系统函数。编成 MATLAB 程序 hc662 如下：

```
syms  z  w0
A=[2*cos(w0), -1;1,0]; B=[1;0];          % 给A,B,C,D赋值
C=[sin(w0),0;cos(w0), -1];D=[0;1];
H=C*inv(z*eye(2) -A)*B+D;                % 调用由状态方程求系统函数的公式
pretty(simple(H))                        % 用美观的形式简化结果
```

程序运行的结果为

$$H=\begin{bmatrix}H_1(z)\\H_2(z)\end{bmatrix}=\begin{bmatrix}\dfrac{Y_1(z)}{U(z)}\\\dfrac{Y_2(z)}{U(z)}\end{bmatrix}=\begin{bmatrix}\dfrac{\sin(w_0)\,z}{z^2-2\,z\cos(w_0)+1}\\\dfrac{z(z-\cos(w_0))}{z^2-2\,z\cos(w_0)+1}\end{bmatrix}\tag{6.6.12}$$

只要把分子分母同乘以 z^{-2}，把其中 z 的正幂换为 z 的负幂，它和信号流图法推导的结果（6.6.11）式完全相同。

6.7　各种结构形式的比较

工程实际中要考虑的问题是多方面的，可以从不同的角度进行比较。本节只从计算复杂性和减小系数量化影响两个方面做一简要的说明。

6.7.1　从计算的复杂性做比较

数字滤波器结构的计算复杂性是由它的乘法器的总数和双输入端加法器的总数来表征的。因为这个数据粗略地表明了滤波器的实现成本和运行时间。这里把本节讨论的各种滤波器的计算复杂性做一个归纳，列于表 6.7.1 中。

表 6.7.1　各种滤波器结构的计算复杂性

滤波器类型 \ 结构类型	M 阶 FIR 滤波器		N 阶 IIR 滤波器(N=M 时)	
	乘法器数目	双端加法器数目	乘法器数目	双端加法器数目
直接型	M+1	M	2N+1	2N
级联型	M+1	M	2N+1	2N
级联格型	2(M+1)	2(M+1)		

（续表）

结构类型 \ 滤波器类型	M 阶 FIR 滤波器		N 阶 IIR 滤波器(N=M 时)	
	乘法器数目	双端加法器数目	乘法器数目	双端加法器数目
线性相位型	$(M+2)/2$	M		
并联型			$2N+1$	$2N$

6.7.2 从减小系数量化影响做比较

系数量化的影响将在第 9 章中详细讨论。这里只给出简明的结论。直接型结构形成的是一个高阶的多项式，高阶多项式的根对于系数的变动非常敏感。系数的微小变动往往可以引起根的很大变化。对于 IIR 滤波器的分母多项式，它的根就是系统的极点，当极点的位置比较靠近单位圆时，其变化对系统的影响尤其严重。系数量化产生微小变动对直接型滤波器特性的影响是相当大的，有时甚至会使滤波器失去稳定性而无法工作，所以设计中应避免采用直接型。二阶级联型结构在减小系数量化影响方面远远优于直接型结构，两者相比，应当优先采用。二阶并联型结构在减小系数量化影响方面性能也很好，它还有处理时间短的优点。其他还有一些受系数量化影响较小的结构，读者可参阅有关文献，不要受本节简要介绍的局限。

复习思考题

6.1 在数字信号处理中的信号流图与连续系统的信号流图相比，有什么共同点？有什么不同点？

6.2 何谓信号流图的环路？何谓环路增益？有环路是不是就意味着有反馈和递归运算，是否一般就说明描述的是 IIR 滤波器？

6.3 何谓代数环？为什么在结构中要避免代数环？

6.4 怎样把信号流图用矩阵方程来表述？其连接矩阵的阶次由什么因素决定？为什么连接矩阵一定是符号矩阵？输入矩阵呢？

6.5 怎样保证连接矩阵是符号矩阵？为什么不必给其中的非零元素逐个赋值？

6.6 解释 FIR 滤波器的几种结构类型的意义，并画出它们的结构特征：直接型；级联型；线性相位型和频率样本型。MATLAB 提供了哪些函数来进行它们之间的参数变换？

6.7 为什么线性相位型结构比直接型结构可以减小一半乘法次数？

6.8 频率样本型的滤波器结构中带有分母部分，为什么它仍然是 FIR 滤波器？

6.9 为什么 IIR 滤波器的直接结构分成 I 型和 II 型？哪一种比较好？为什么？

6.10 什么是 IIR 滤波器的级联结构，解释由它的直接结构转换为级联结构的主要步骤。说明 tf2sos 函数的输入输出变元的含义。

6.11 画出信号通过级联型滤波器的框图，解释其计算程序 sosfilt 的主要流程。

6.12 什么是 IIR 滤波器的并联结构，解释由它的直接结构转换为并联结构的主要步骤。说明 tf2par 函数的输入输出变元的含义。

6.13 画出信号通过并联型滤波器的框图，解释其计算程序 parfilt 的主要流程。

6.14 根据例 6.4.3 的计算结果，写出该滤波器的传递函数的三种形式。

6.15 什么是格型滤波器？什么是格型梯形滤波器？MATLAB 提供了什么函数帮助由直

接型的参数求格型梯形滤波器的参数？

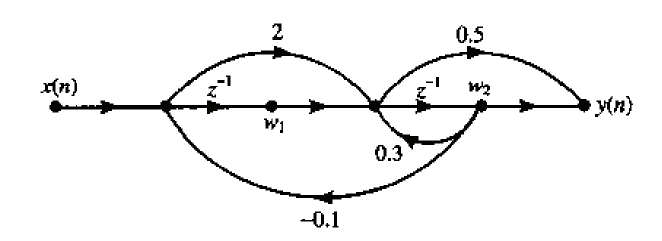
6.16 什么是系统的状态方程？数字系统的状态变量与信号流图中的节点变量有何区别？应该如何选择状态变量？

6.17 如何由状态方程组(6.6.4)式和 (6.6.5)式导出系统传递函数表达式(6.6.7)？

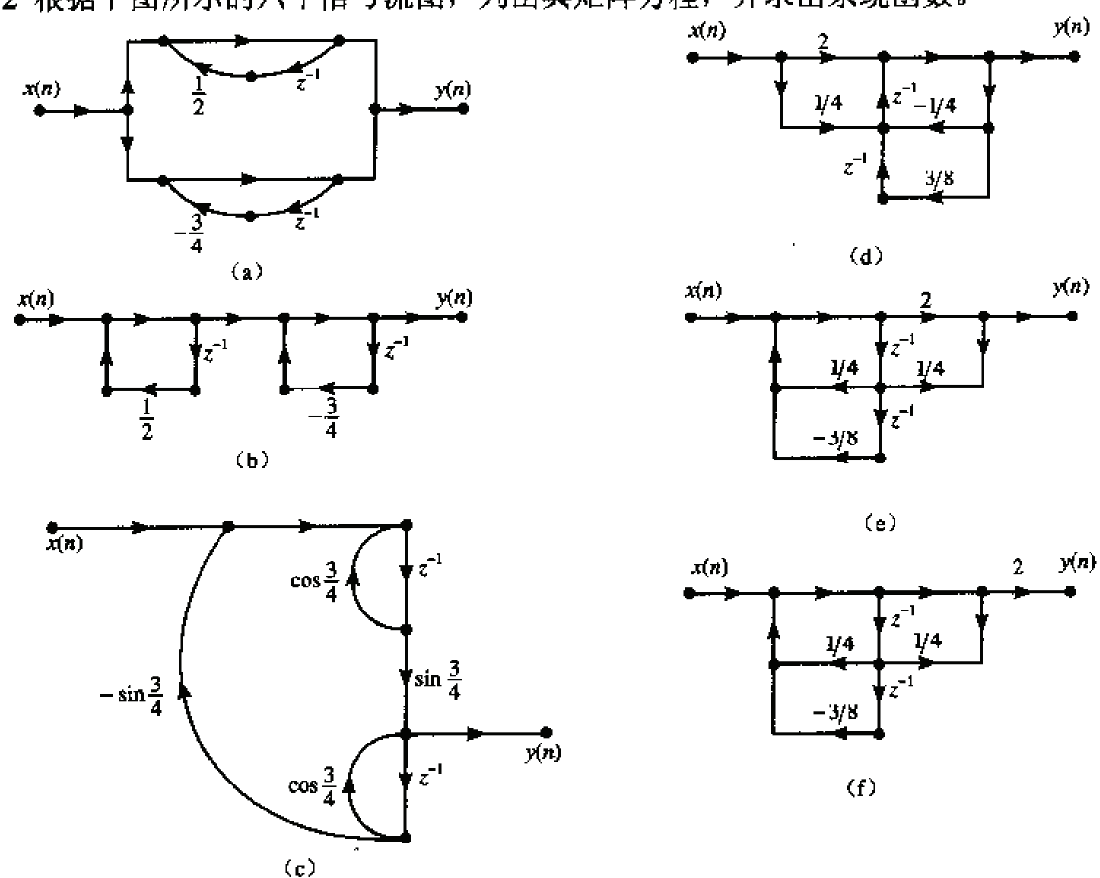
6.18 信号流图和状态方程都是用来表示线性系统的工具，它们之间有什么相同点？有什么不同点？

习题

6.1 根据下图所示的信号流图，列出其矩阵方程，并求出系统函数。



6.2 根据下图所示的六个信号流图，列出其矩阵方程，并求出系统函数。



6.3 已知 FIR 滤波器由下面的差分方程描述，画出下列结构的信号流图并确定其系数。

$$y(n)=\sum_{k=0}^{10}\left(\frac{1}{2}\right)^{|5-k|}x(n-k)$$

(a)直接形式　(b)线性相位形式　(c)级联形式　(d)频率样本形式

6.4 一个 IIR 滤波器由下面的系统函数描述，

$$H(z)=5\left(\frac{1+z^{-2}}{1-0.8z^{-1}+0.64z^{-2}}\right)\left(\frac{2-z^{-1}}{1-0.75z^{-1}}\right)\left(\frac{1+2z^{-1}+z^{-2}}{1+0.81z^{-2}}\right)$$

确定并画出下列结构。

(a)直接 I 型　　(b)直接 II 型

(c)包含二阶直接 II 型子系统的级联形式　(d)包含二阶直接 II 型子系统的并联形式

6.5 一个 IIR 滤波器由下面的系统函数描述：

$$H(z)=\left(\frac{-14.75-12.9z^{-1}}{1-(7/8)z^{-1}+(3/32)z^{-2}}\right)+\left(\frac{24.5+26.82z^{-1}+z^{-2}}{1-z^{-1}+(1/2)z^{-2}}\right)$$

确定并画出下列结构。

(a)直接 I 型　　(b)直接 II 型

(c)包含二阶直接 II 型子系统的级联形式　(d)包含二阶直接 II 型子系统的并联形式

6.6 已知 FIR 滤波器的 16 个频率样本值为

$H(0)=12, H(1)=-3-j\sqrt{3}$，$H(2)=1+j$，$H(3\sim13)=0$，$H(14)=1-j$，$H(15)=-3+j\sqrt{3}$，

试画出其频率采样结构，选择 $r=1$。

6.7 已知(6.3.16)式：$H(z)=\frac{1-z^{-N}}{N}\sum_{k=0}^{N-1}\frac{H(k)}{1-W_N^{-k}z^{-1}}$，试推导证明本书中的(6.3.17) 式和(6.3.18)式。

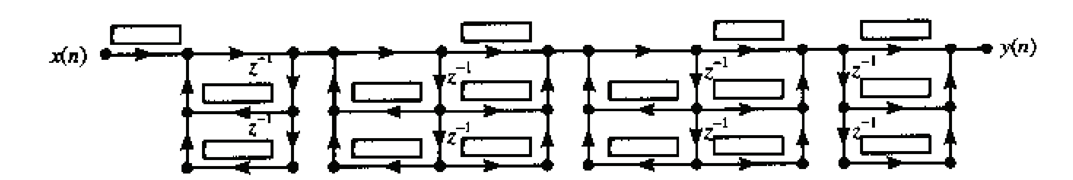
6.8 一个因果、线性、时不变系统为 $y(n)=\sum_{k=0}^{5}\left(\frac{1}{2}\right)^k x(n-k)-\sum_{l=0}^{5}\left(\frac{1}{3}\right)^l y(n-l)$，确定并画出下列结构的方框图。用下面每种情况下的结构，计算系统对 x(n)=u(n), 0≤n≤20 的响应。

(a)直接型　(b)包含二阶直接 II 型子系统的级联形式　(c)格型梯形形式

6.9 一个线性时不变系统具有系统函数如下：

$$H(z)=\frac{0.5(1+z^{-1})^6}{\left(1-\frac{3}{2}z^{-1}+\frac{7}{8}z^{-2}-\frac{13}{16}z^{-3}-\frac{1}{8}z^{-4}-\frac{11}{32}z^{-5}+\frac{7}{16}z^{-6}\right)}$$

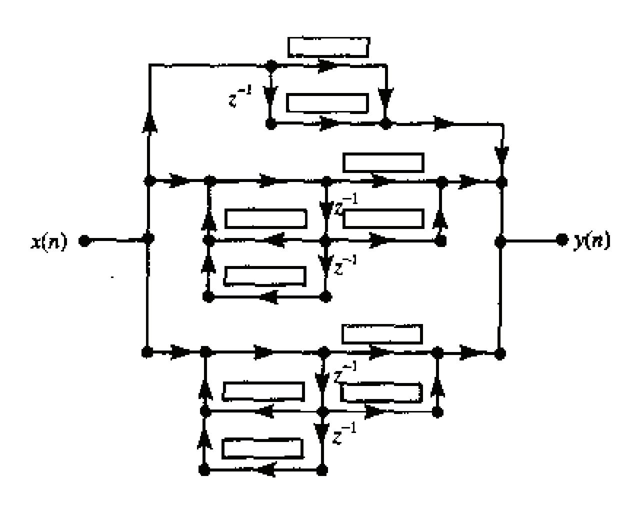
它由下图所示的信号流图实现。(a) 填出方框图中的系数 (b) 这个解惟一吗？请说明。



6.10 一个线性时不变系统具有系统函数如下：

$$H(z)=\frac{5+11.2z^{-1}+5.44z^{-2}-0.384z^{-3}-2.35z^{-4}-1.29z^{-5}}{1+0.8z^{-1}-0.51z^{-3}-0.41z^{-4}}$$

它由下图所示的信号流图实现。请填出方框图中的系数。



6.11 以级联形式实现具有以下脉冲响应的 IIR 滤波器($n \geq 0$)。

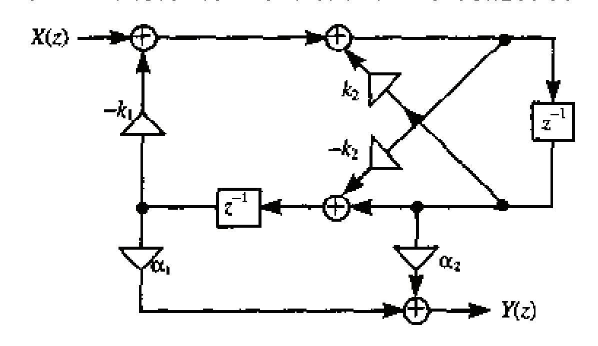
$$h(n) = (0.2)^n + 5(-0.6)^n + (0.9)^n \sin(0.3n)$$

6.12 根据题 6.4 和题 6.5 所给的系统函数，确定、画出格型梯形结构，并确定其参数。

6.13 一个 FIR 滤波器的格型参数为：K_0=2,K_1=0.6,K_2=0.3,K_3=0.5,K_4=0.9。求出它的脉冲响应，并画出上述滤波器的直接形式和格型形式结构。

6.14 用状态方程表示题6.1的信号流图，并用状态方程法导出其系统函数。

6.15 下图所示滤波器结构，其中所有的乘子均为实系数。（a）求出其传递函数 $H(z)=Y(z)/X(z)$；（b）讨论这些乘数在什么范围内时，系统能保持BIBO稳定性。



第7章　FIR 滤波器设计

前面几章讨论了给定系统特性和输入序列，求取系统在时域或频域输出的问题，这是信号处理中的正问题。现在我们来研究逆问题，那就是给定了输入输出指标，设计能满足这些指标的系统。这种可以人为调整参数来满足给定要求的硬件或软件系统，称为滤波器。这是更重要、更困难的一个问题，这种问题没有惟一的解，因为可能有多种类型的滤波器都能满足给定的要求。要人为地进行选择，就要考虑更多的因素，所以设计比分析更加接近工程实际。在学校的教学中，只能着重于设计的基本原理方面。

前面指出，有两种类型的数字滤波系统。一种是 FIR 数字滤波器；另一种是 IIR 数字滤波器。在本章和下一章中，将讨论 FIR 和 IIR 滤波器的几种基本设计算法。这些设计大多数是针对选频型的，也就是说，主要针对频率特性的要求来进行设计。从幅频特性的要求而言，有低通、高通、带通和带阻滤波器；从相频特性来说，往往要求线性相位。在 FIR 滤波器设计中，还会涉及像微分器或希尔伯特变换器之类的系统，尽管它们不是选频滤波器，但是也遵循选频滤波器的设计技术。目前，读者可能还不知道这两类系统有什么用途，但为了理论上的完整性，本章也将做最低限度的介绍。

更完善的滤波器设计可以基于任意频域指标。这是一门专门学问，有专门的著作，例如参考文献[7]。但这些问题已超出了本书的范围，因为滤波器设计在相当程度上是一个工程问题，在计算技术高度发展的今天，几乎所有的电子设计软件中都把滤波器设计作为其模块之一。那种用查表、手工计算的方法在工程实践中已经濒于淘汰。为了让读者也了解这些情况，除了在讲解滤波器设计中充分调用 MATLAB 函数之外，还在第 8 章讲完 IIR 滤波器之后，将对 MATLAB 中的滤波器设计工具作一介绍。

7.1　数字滤波器设计的基本要求

7.1.1 数字滤波器的设计基本步骤

数字滤波器的设计要经过三个步骤：

（1）确定指标：在设计一个滤波器前，必须有一些指标。这些指标要根据应用确定。

（2）模型逼近：一旦确定了技术指标，就可利用前面学习过的基本原理和关系式，提出一个滤波器模型来逼近给定的指标体系。这是滤波器设计所要研究的主要问题。

（3）实现：上面两步的结果得到的滤波器，通常是以差分方程、系统函数或脉冲响应来描述的。根据这个描述，可以如第 6 章所讨论的那样，用硬件或计算机软件来实现它。在本章及下一章中，我们将详细讨论第二步，即如何把指标转换成一个滤波器模型。

7.1.2　数字滤波器的主要指标

在很多实际应用，如语音或音频信号的处理中，数字滤波器用来实现选频功能。因此，

指标的形式应为频域中的幅度和相位响应。在通带中，通常期望具有线性相位响应。正如在第 5 章中讨论的，在 FIR 滤波器中，可以得到精确的线性相位。在 IIR 滤波器中，通带中的线性相位是不可能得到的，因此，主要考虑幅度指标。

幅度指标可以两种方式给出。第一种，叫做绝对指标，它提出了对幅度响应函数$|H(\mathrm{j}\omega)|$的要求。这些指标一般可直接用于 FIR 滤波器。IIR 滤波器也可用这些指标，但因为 IIR 滤波器的设计是由模拟滤波器转换而来，模拟滤波器通常采用平方幅度指标。所以中间要经过一次转换，我们将在第 8 章中讨论它。第二种方法叫做相对指标，它以分贝（dB）值的形式提出要求，其值定义为：

$$\text{dB值} = -20\log_{10}\frac{|H(\mathrm{e}^{\mathrm{j}\omega})|}{|H(\mathrm{e}^{\mathrm{j}\omega})|_{\max}} \geqslant 0 \tag{7.1.1}$$

经过定义中所包含的归一化，所有滤波器的相对幅频特性最高处的值为 0 dB。由于定义(7.1.1)式中有一个负号，幅频特性小的地方，其 dB 值反而是正的。在实际中，这种方法最受欢迎。为了说明这些指标，下面以低通滤波器设计为例进行讨论。

绝对指标：低通滤波器的典型绝对指标如图 7.1.1(a)所示。其中

(1) $[0,\omega_p]$段叫做通带，δ_1是在理想通带中能接受的振幅波动（或容限）；

(2) $[\omega_s,\pi]$段叫做阻带，δ_2是阻带中能接受的振幅波动（或容限）；

(3) $[\omega_p,\omega_s]$叫做过渡带，在此段上对幅度响应通常没有限制，也可以给些弱限制。

相对指标（dB）：低通滤波器的典型相对技术指标如图 7.1.1(b)所示。其中三种频带的定义不变，只是

(1) R_p是通带波动的 dB 值；

(2) A_s是阻带衰减的 dB 值。

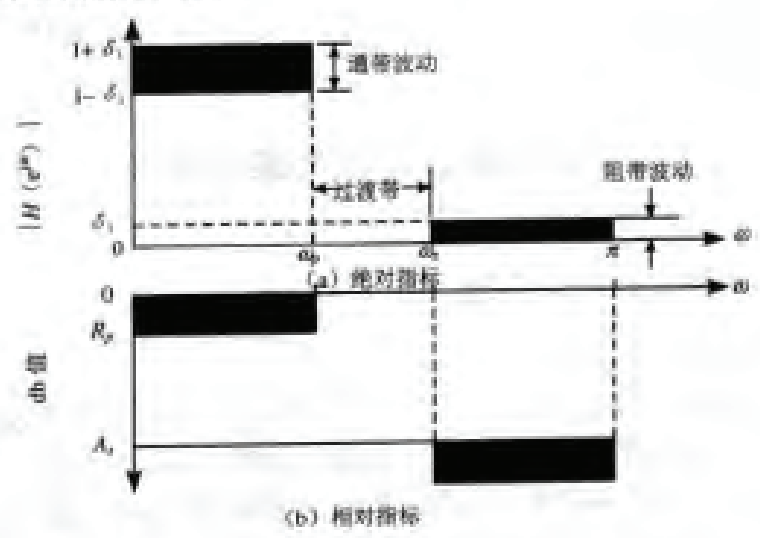


图 7.1.1　FIR 滤波器技术指标

上述两种指标存在一定的数学关系。由于绝对指标中的$|H(\mathrm{j}\omega)|_{\max}=(1+\delta_1)$，因此，

$$R_P = -20\log_{10}\frac{1-\delta_1}{1+\delta_1} > 0 \tag{7.1.2}$$

$$A_s = -20\log_{10}\frac{\delta_2}{1+\delta_1} >> 1 \tag{7.1.3}$$

逆向的关系为

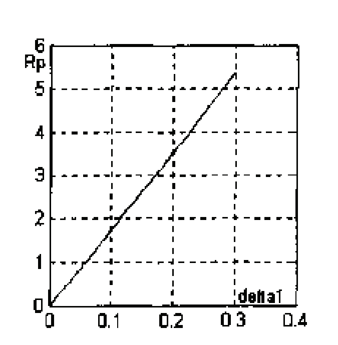
$$\delta_1 = \frac{1-10^{-\frac{R_p}{20}}}{1+10^{-\frac{R_p}{20}}} \tag{7.1.4}$$

$$\delta_2 = (1+\delta_1)10^{-\frac{As}{20}} \approx 10^{-\frac{As}{20}} \tag{7.1.5}$$

可以把(7.1.2)式、(7.1.3)式和(7.1.4)式、(7.1.5)式用 MATLAB 语句描述如下(见 fgp711)：

```
Rp=-20*log10((1-delta1)./(1+delta1));       % 由delta1 及delta2 求Rp 的语句
As=-20*log10(delta2./(1+delta1));           % 由delta1 及delta2 求As 的语句
delta1=(1-10^(-Rp/20))./(1+10^(-Rp/20))     % 由Rp 求delta1 的语句
delta2=(1+delta1).*10^(-As/20)              % 由As 及delta1 求delta2 的语句
```

用这些程序语句，可以准确地求出滤波器绝对指标和相对指标的对应数值。利用（7.1.2）和（7.1.3）式，也可以画出如图 7.1.2 所示曲线，以便快速粗略地正反查找。



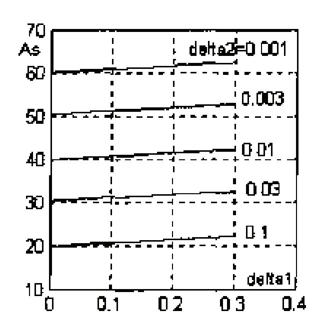


图 7.1.2　滤波器绝对和相对指标之间的数据曲线

例 7.1.1：设一滤波器技术指标如下，通带波动为 1 dB，阻带衰减为 50 dB。试确定 δ_1和 δ_2。

解：利用图 7.1.2，可大致查出 delta1=0.055,delta2=0.003。而用（7.1.4）、（7.1.5）式得到的精确值为 delta1 = 0.0575，delta2 = 0.0033。

相位指标主要是线性相位条件。即理想的滤波器相位 θ 应该和频率 ω 成正比，也就是

$$\theta(\omega) = -\tau\omega \tag{7.1.6}$$

如果频率为 ω 的信号迟延的相角为 θ，则对应的时间延迟为 θ/ω。因此满足（7.1.6）式的系统对信号中所有的频率分量都具有相同的时间延迟 τ（其单位为拍数，没有量纲），如果该滤波器的振幅特性又是常数，则通过该滤波器的信号仅仅有一个纯粹的时延，而不会产生波形失真。这在绝大多数的应用场合下都是很受欢迎的特性。略低一点的要求是相位 θ 和频率 ω 成线性关系，即

$$\theta(\omega) = \theta_0 - \tau\omega \tag{7.1.7}$$

满足这个条件的系统不能保证对信号中所有的频率分量都具有相同的时间延迟，因为 $\theta/\omega=\theta_0/\omega-\tau$ 不是常数，多了一个非常数项 θ_0/ω，因此信号通过这样的系统仍然会产生相位失真。不过如果 θ_0 比较小，θ_0/ω 比 τ 小得多，那么它的失真往往仍可以忽略不计。例如下一节要讨论的类型 III 和类型 IV 线性相位滤波器，它们的 θ_0 都为 $\pi/2$，而 τ 的值是 FIR 滤波器长度的一半，通常可以到几十、甚至几百，所以 θ_0 的影响通常可以忽略不计。习惯上常说的是对 FIR 滤波器有线性相位要求，而没有特别强调为正比相位要求。

对于更多的滤波器，特别是 IIR 滤波器，线性相位是无法做到的。这时衡量它们造成的相位失真常用群迟延 $g(\omega)$来描述。群迟延定义为相位 θ 对频率 ω 的导数：$g(\omega)=\dfrac{\mathrm{d}\theta(\omega)}{\mathrm{d}\omega}$。

对于前面所说的两种线性相位滤波器而言，其群迟延 g 都等于常数 τ。对不符合线性相位的滤波器，则应要求其群迟延 g 随 ω 变化的特性尽量平，越接近于常数越好。

7.1.3 为什么以低通滤波器为典型

上述指标是针对低通滤波器的，其他类型的选频滤波器，像高通和带通滤波器，它们的技术指标可作类似的规定，并且可以由低通滤波器演化而得。因此，在讨论设计技术时，将集中讨论低通滤波器，最重要的设计参数是通带波动和边缘频率。在下一章中讨论怎样把一个低通滤波器转换成其他类型的选频滤波器。因此重点要搞清低通滤波器的设计技术。

设计一个低通滤波器，也就是，找到它的系统函数 $H(z)$或差分方程，使它的通带为$[0,\omega_p]$，容限为 δ_1（或 R_p），阻带为$[\omega_s,\pi]$，容限为 δ_2（或 A_s）。

在本章中，将讨论 FIR 数字滤波器的设计和逼近。FIR 滤波器在设计和实现上具有如下的优越性：

(1) 相位响应可为严格的线性，因此它不存在延迟失真，只有固定的时间延迟。

(2) 由于不存在稳定性问题，所以设计相对简单。

(3) 只包含实数算法，不涉及复数算法，不需要递推运算，长度为 M 的滤波器（阶数为 $M-1$），它的计算值约为 $M/2$。

首先讨论线性相位 FIR 滤波器的性质。这是设计算法中很重要的基础，几乎所有的 FIR 滤波器都满足线性相位条件。然后研究三种设计算法，分别为窗函数设计技术，频率样本设计技术和等波动设计技术。

7.2 线性相位滤波器的特性

在第 5.5.2 节中研究了线性相位条件。分析了它对 FIR 滤波器的脉冲响应 $h(n)$ 的要求，也就是要求滤波器分子系数 b_n 满足对称性条件。如果滤波器的系数 $h(n)$的长度为 N，（或滤波器的阶次为 $M=N-1$），则这些系数应关于中心点 $\tau=n0=(N-1)/2$ 对称。偶对称时，$h(n)=h(N-n-1)$；而奇对称时，$h(n)=-h(N-n-1)$。再考虑到 N 可以为奇数或偶数，总共有四种类型的线性相位 FIR 滤波器。下面讨论这四种滤波器频率响应的计算方法。

7.2.1 线性相位和符幅特性(Amplitude Response)

以前讨论频率响应时，幅频特性通常用$|H(\omega)|$表示，$H(\omega)$是复数变量，在复平面上可以看做向量。则$|H(\omega)|$是向量 $H(\omega)$的长度，因此永远取正值。其实向量的幅度是可以取成负的，例 3.2.3 中的图 3.2.2 就显示了幅频特性有负的部分。幅特性的反号可以用相频特性上加（或减）π 来补偿。所以幅特性常为正的概念是一种误解，在人们不关心相特性的时候，可以让它常取正值，而靠改变相特性来补偿。当相特性非常重要，不容许随便增减时，幅特性就必须分出正负来。线性相位滤波器强调了相位的线性，不允许随便增减相位，因此就提出了幅特性符号的问题。在英文中，不考虑正负的幅特性称为 magnitude response，而考虑正负号时称为 amplitude response。中文的信号处理书中，作者还未见到有将这两者区分的译名。往往都是含糊对待，都称为幅频特性，这对读者弄清概念不利。因此作者在此启用了“符幅特性”这个名词表示具有正负号的幅特性，用 $A(\omega)$表示，以区别于幅特性$|H(\omega)|$。

下面分别讨论 5.5.2 中提出的四种线性相位滤波器的符幅特性计算方法：

(1)类型 I，系数对称，N 奇数。过去求幅特性的习惯做法是简单地求频率特性的模值，在 MATLAB 中就是用 abs(H)函数，为了计算符幅特性，这种方法就不能用了，只能直接作解析计算，求出频率特性的解析式，并将它分解为用幅度和相位来表述，即 $H(\omega)=A(\omega)e^{j\theta(\omega)}$，其中 $\theta(\omega)$应满足线性相位条件，这时的 $A(\omega)$就是符幅特性。对于长度为 N 的序列 $h(n)$，先考虑两个相等（对称）的系数 $h(n)$和 $h(N-1-n)$，这两个样本的傅里叶变换为

$$\begin{aligned}H_n(e^{j\omega})&=h(n)e^{-j\omega n}+h(N-1-n)e^{-j\omega(N-1-n)}=h(n)[e^{-j\omega n}+e^{-j\omega(N-1-n)}]\\&=h(n)e^{-j\omega(N-1)/2}[e^{-j\omega n}e^{j\omega(N-1)/2}+e^{j\omega n}e^{-j\omega(N-1)/2}]\\&=h(n)e^{-j\omega(N-1)/2}\cdot 2\cos[\omega((N-1)/2-n)]=A_n(\omega)e^{-j\omega(N-1)/2}\end{aligned}\quad(7.2.1)$$

序列 h 的下标 n 从 0 起算，中点位置是$(N-1)/2$ 到 $N-1$ 为止。因为(7.2.1)式取的是成对的系数样本，为了避免重复，(7.2.1)式中的 n 的取值范围为 $n=0:(N-1)/2-1$（不能包含中点）。为了使四种类型滤波器的公式简明统一，引进两个参数。一个是令 $\tau=(N-1)/2$，它就是(7.1.5)式中的 τ 值，用以反映延迟的样本数或群延迟，所以它可以是分数。另一个是 $L=\lfloor(N-1)/2\rfloor$，即把 τ 值向下取整，L 必须是整数，只有整数才能用做下标。当 N 为奇数时，$\tau=L$。讨论到类型 II 时，将会看到区别 τ 与 L 的道理。

因此(7.2.1)式适用于 $n=0,1,2,\ldots,L-1$。在计算整个序列 $h(n)$的符幅特性时，共有 L 对对称的系数[从 $n=0$ 到 $n=L-1$，再加上对称中心 $L=(N-2)/2$ 处的系数样本 $h((L)$单独引起的符幅特性。也可以把这个单独的 $h((L)$算到前面成对系数向量的求和号下，然后在后面减掉一个 $h(L)$。所以有 L+1 个如(7.2.1)式所表示的成对项，减一个单独项，可以得出公式(7.2.2 a)。

$$H(e^{j\omega})=\left[\sum_{n=0}^{L}2h(n)\cdot\cos(\omega(\tau-n))-h(L)\right]e^{-j\omega\tau}=A(\omega)e^{-j\omega\tau}\quad(7.2.2\text{ a})$$

这样做的好处是四种类型滤波器的计算公式比较统一，便于编程。

很明显，$e^{-j\omega\tau}$ 代表相角项，相角为 $\theta(\omega)=-\omega\tau$,它与 ω 成严格的线性关系；$A(\omega)$就是符幅项。因此，可以写成（7.2.2b）式。

$$A(\omega)=\sum_{n=0}^{L}2h(n)\cdot\cos[\omega(\tau-n)]-h(L)\qquad \omega=[\,0\sim2\pi]\quad(7.2.2\text{ b})$$

可以看出，符幅特性是由 L+1 项叠加组成的。n=0 时的符幅项为 $2h(0)\cos(L\omega)$，它的余弦的频率最高，在奈奎斯特频率区间 $\omega=[0\sim2\pi]$之间波动 L 个周期；n=L-1 时，对应的余弦分量为 $2h(L-1)\cos(\omega)$，在 $\omega=[0\sim2\pi]$之间波动一周。L 说明了符幅特性波动的最高阶次。由于余弦是偶函数，关于 $\omega=\pi$ 是对称的，所以可以只研究在 $\omega=[0\sim\pi]$的一段。余弦函数在 $\omega=0$，π 和 2π 处都不等于零，因此类型 I 线性相位滤波器既可以用做低通滤波器(在 $\omega=0$ 处，幅特性不为零)；也可以用做高通滤波器(在 $\omega=\pi$ 处，幅特性不为零)；而且也可以用做带通和带阻滤波器，故其应用最为广泛。

(2)类型 II，系数对称，N 偶数。如果 N 为偶数，那么全部系数都可以配对，不会出现中心点的单项，一共有 $N/2$ 组数值相同对称出现的系数，(7.2.2a)式就成为

$$H(e^{j\omega})=\left[\sum_{n=0}^{N/2-1}2h(n)\cdot\cos[\omega((N-1)/2-n)]\right]e^{-j\omega(N-1)/2}=A(\omega)e^{-j\omega(N-1)/2}\quad(7.2.3\text{a})$$

此时 $\tau=(N-1)/2$ 将不是整数，也就是说，对称中心将在两个样本点的中间。其相角特性

仍为 $\theta(\omega)=-\omega\tau$,它仍与 ω 成严格的线性关系。为了和类型 I 用相似的算式，此时用整数 $L=\lfloor (N-1)/2\rfloor=N/2-1$ 来表示求和号的上限。得出其符幅特性的表达式(7.2.3b)。

$$A(\omega)=\sum_{n=0}^{L}2h(n)\cdot\cos[\omega(\tau-n)] \qquad \omega=[\,0\sim2\pi\,] \tag{7.2.3b}$$

可见，它的符幅特性也由 $L+1$ 个余弦分量构成。这样，不管 N 为奇数或偶数时，L 都是整数，求和的项数都是 $L+1$。当 N 是偶数时，由于 $\tau=(N-1)/2$ 是分数，这个求和号中的所有余弦分量 $\cos((\tau\ n)\omega)$ 中都含一个 0.5ω。在 $\omega=\pi$ 处，就会出现 $\cos((\tau-n)\pi)=\cos((N/2-n-0.5)\pi)=0$, 因此类型 II 线性相位滤波器的符幅特性在 $\omega=\pi$ 处必定为零。这样的幅特性当然不能用做高通和带阻滤波器。由于余弦波形关于其过零点奇对称，所以类型 II 的线性相位滤波器符幅特性关于 $\omega=\pi$ 呈奇对称。

(3)类型 III，系数反对称，N 奇数。 当系数序列反对称时，在(7.2.1)式的推导过程中，$h(N-1-n)$ 项将反号而有 $h(N-1-n)=-h(n)$，这将导致(7.2.1)式中的 $\dfrac{e^{jx}+e^{-jx}}{2}=\cos x$ 成为 $\dfrac{e^{jx}-e^{-jx}}{2}=j\sin x=e^{j\pi/2}\sin x$。所以要把公式(7.2.2) 和(7.2.2a)中做如下改动：一是在符幅项中把 cos 换成 sin，二是在相角项中增加一个 $\pi/2$。结果为

$$\begin{aligned}H(e^{j\omega})&=\left[\sum_{n=0}^{L-1}2h(n)\cdot\sin(\omega((N-1)/2-n))+h(L)\right]e^{j(\pi/2-\omega(N-1)/2)}\\&=A(\omega)e^{j(\pi/2-\omega(N-1)/2)}\end{aligned} \tag{7.2.4 a}$$

当 N 为奇数时，$\tau=L=$整数。相角特性为 $\theta(\omega)=\pi/2-\omega(N-1)/2=\pi/2-\omega\tau$,它虽然仍与 ω 成线性关系，但多了一个常数项 $\pi/2$，故 $\omega=0$ 时，$\theta\neq0$，相特性不通过原点。注意反对称序列的对称中心 $n=L$ 处，序列的值 $h(L)=0$。所以得知它的符幅特性为

$$A(\omega)=\sum_{n=0}^{L}2h(n)\cdot\sin[\,\omega(\tau-n)\,] \qquad \omega=[\,0\sim2\pi\,] \tag{7.2.4b}$$

它是由 L 对反对称系数组成的分量为基础的符幅特性的代数和。可以看出，总的符幅特性是由 L 个频率按整数倍增加的正弦波叠加组成的，因为 $h(L)=0$，把这项加上没有关系，所以求和号中仍含 $L+1$ 项。$n=0$ 时的符幅项为 $2h(0)\sin(\tau\omega)$，它随 ω 做正弦摆动的频率最高，在奈奎斯特频率范围 $\omega=[0\sim2\pi]$之间波动 L 个周期；$n=L-1$ 时，对应的正弦分量为 $2h(L-1)\sin(\omega)$，在 $\omega=[\,0\sim2\pi]$之间波动一周。L 仍表示符幅特性波动的最高阶次。

由于正弦是奇函数，对于 $\omega=0$ 和 π 都是反对称的，所以也可以只研究在 $\omega=[0\sim\pi]$的一段符幅特性。正弦函数在 $\omega=0$，π 和 2π 处都等于零，因此类型 III 线性相位滤波器的符幅特性在 $\omega=0$，π 和 2π 处都等于零。它既不能用做低通滤波器；也不能用做高通滤波器，更不能用做带阻滤波器，只能用做带通滤波器(在 $\omega=0$ 和 π 处，幅特性为零)；类型 III 的滤波器一般不用于选频滤波器。

(4)类型 IV，系数反对称，N 偶数。 根据与类型 II 滤波器同样的道理，如果 N 为偶数，那么就不出现(7.2.4a)式中的单项，一共有 $N/2$ 组数值相同反对称出现的系数，(7.2.4a)式就成为

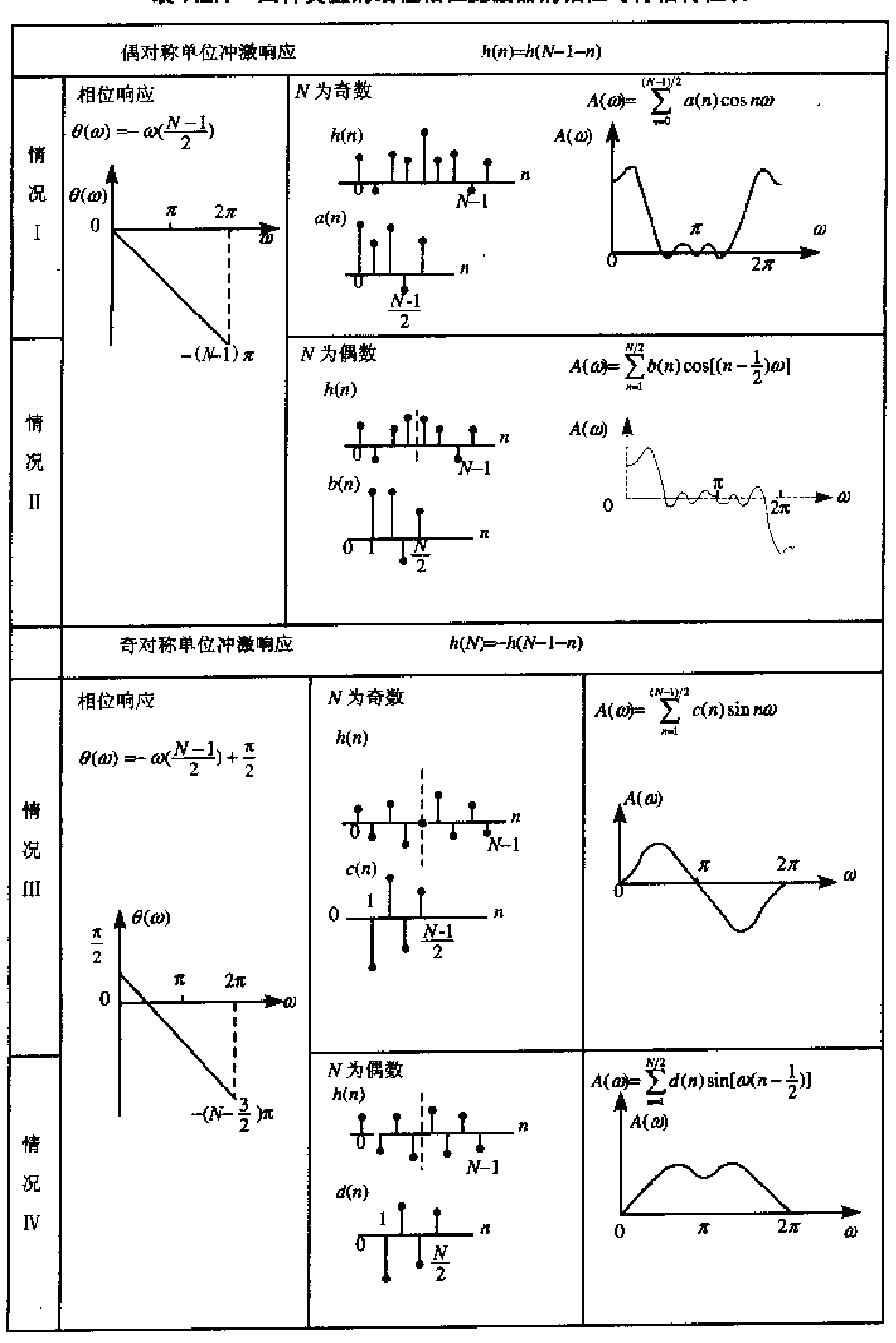
$$H(e^{j\omega})=\left[\sum_{n=0}^{L}2h(n)\bullet\sin[\omega(\tau-n)]\right]e^{-j\omega\tau}=A(\omega)e^{-j(\pi/2-\omega\tau)} \tag{7.2.5a}$$

此处仍有 $\tau=(N-1)/2$，因此 τ 将不是整数,带有小数部分 0.5。也就是说，对称中心将在两

个样本点的中间。它的相角特性为 $\theta(\omega)=\pi/2-\omega\tau$，虽然仍与 ω 成线性关系，但多了一个常数项 $\pi/2$，故不通过原点。其总的符幅特性为

$$A(\omega)=\left[\sum_{n=0}^{L}2h(n)\cdot\sin[\omega(\tau-n)]\right]\qquad \omega=[0\sim2\pi]\qquad (7.2.5\ \mathrm{b})$$

表 7.2.1　四种类型的线性相位滤波器的相位与符幅特性表



此时因为 L 是整数，它的符幅特性也由 L+1 个正弦分量构成。因为 sin(0)=0，在 ω=0 处，符幅特性必定为 $A(0)=0$。又因为 τ 不是整数,这个求和号中的所有正弦分量 $\sin[(\tau-n)\omega]$中都含有（整数+0.5）ω 项，于是在 $\omega=\pi$ 处的符幅特性 $A(\pi)$不必为零；类型 IV 的线性相位滤波器的符幅特性在 ω=0 处为零，这样的特性显然不能用做低通和带阻滤波器。实际上这种类型的滤波器也不适用于选频滤波。

把这四种类型的线性相位滤波器的相角特性和符幅特性用对比的方法画在一张图上如表 7.2.1。其中最左的一列为相频特性；中间一列为序列的示意图，注意它的对称性；这两列的特点比较简单和直观，在推导公式中已经可以看清楚了。最右列是符幅特性，从图中可以看出它的阶次（波动次数）和对称性。至于应该如何选用，将在以后讨论。

7.2.2　符幅特性的计算举例

例 7.2.1：设系统的系数向量为 b=[-1,2,4,2,-1]，a=1，分别求其幅频特性及符幅特性，并作比较。

解：先用手工计算，因为 a=1,说明这是一个 FIR 滤波器，观察系数向量 b，其长度 N=5，阶次为 $M=N-1=4$，传递函数为

$$H(z)=-1+2z^{-1}+4z^{-2}+2z^{-3}-z^{-4}$$

注意：$h(n)=b_n$，其频率响应为

$$H(\mathrm{e}^{\mathrm{j}\omega})=\sum_{n=0}^{4}h(n)\mathrm{e}^{-\mathrm{j}\omega n}=-1+2\mathrm{e}^{-\mathrm{j}\omega}+4\mathrm{e}^{-\mathrm{j}2\omega}+2\mathrm{e}^{-\mathrm{j}3\omega}-\mathrm{e}^{-\mathrm{j}4\omega} \tag{7.2.6}$$

按照一般求频率特性的方法，先把各项的实部和虚部分开，相加后再合成，则有

$$H(\mathrm{e}^{\mathrm{j}\omega})=\sum_{n=0}^{4}h(n)\cos(n\omega)/2+\mathrm{j}\bullet\sum_{n=0}^{4}h(n)\sin(n\omega) \tag{7.2.7}$$

前面的求和号为 H 的实部 Re(H)，后面的求和号为 H 的虚部 Im(H)。给定一个 ω，可以求出 H 的实部和虚部；给出一系列 ω构成的数组，就可得到一个 $H(\omega)$数组；这是个复数数组，其幅特性为

$$\left|H(\mathrm{e}^{\mathrm{j}\omega})\right|=\sqrt{\mathrm{Re}[H(\omega)]^2+\mathrm{Im}[H(\omega)]^2} \tag{7.2.8}$$

相特性为

$$\angle H(\mathrm{e}^{\mathrm{j}\omega})=\arctan[\mathrm{Im}[H(\omega)/\mathrm{Re}[H(\omega)]] \tag{7.2.9}$$

虽然这是很简单的题目，但要靠手工算十来个点的值再把它相加也要费不少工夫。好在有 MATLAB 的 H=freqz(b,1,N)可以调用。不管用何种方法，由(7.2.8)式求出的幅频特性永远只会出现正数，不会出现负数。

下面来看符幅特性的计算方法。将 n=0,4 组成一对，n=1,3 组成一对，n=2 处是对称中心，$h(2)$只能独立计算，于是有

$$\begin{aligned}H(e^{\mathrm{j}\omega})&=-\mathrm{e}^{-\mathrm{j}2\omega}(\mathrm{e}^{\mathrm{j}2\omega}+\mathrm{e}^{-\mathrm{j}2\omega})+2\mathrm{e}^{-\mathrm{j}2\omega}(\mathrm{e}^{\mathrm{j}\omega}+e^{-\mathrm{j}\omega})+4\mathrm{e}^{-\mathrm{j}2\omega}\\&=-2\cos(2\omega)\mathrm{e}^{-\mathrm{j}2\omega}+4\cos(\omega)\mathrm{e}^{-\mathrm{j}2\omega}+4\mathrm{e}^{-\mathrm{j}2\omega}=A(\omega)\mathrm{e}^{-\mathrm{j}2\omega}\end{aligned} \tag{7.2.10}$$

可见，由于对称配置，$\mathrm{e}^{-\mathrm{j}n\omega}$ 的高次和低次项搭配成为余弦，合成后的相位项都是$\mathrm{e}^{-\mathrm{j}2\omega}$，故其符幅特性是这三个幅度项的代数和：

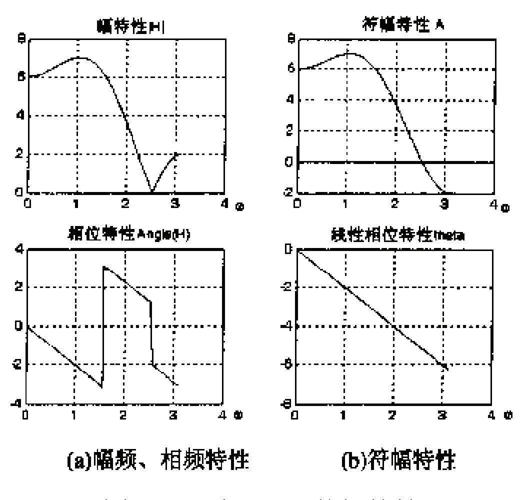
$$A(\omega)=-2\cos(2\omega)+4\cos(\omega)+4 \tag{7.2.11}$$

$$\theta(\omega)=-2\omega \tag{7.2.12}$$

可见，相角 θ 与 ω 成严格的正比关系，所以相应的 $A(\omega)$ 就是符幅特性。现在用 MATLAB 程序来计算其数值结果并做比较，写出程序 hc721 如下：

```
b=[-1,2,4,2, -1];a=1;                          % 给定滤波器系数向量
[H,w]=freqz(b,a);                              % 求频率特性
subplot(2,2,1),plot(w,abs(H))                  % 求并画出幅频特性
subplot(2,2,3),plot(w,angle(H))                % 求并画出幅频特性
A=-2*cos(2*w)+4*cos(w)+4;                      % 求符幅特性
theta=-2*w;                            % 求线性相位特性
subplot(2,2,2),plot(w,A)                       % 画出符幅特性
subplot(2,2,4),plot(w,theta)                   % 画出线性相位特性
```

程序的运行结果画在图 7.2.1 中。



(a)幅频、相频特性　(b)符幅特性

图 7.2.1　例 7.2.1 的幅特性

可以看出，在 ω 约为 2.5 处，幅特性有一个过零点。从符幅特性曲线上看，它是连续而光滑地由正值变为了负值；而在幅频特性上，它急剧地折向正方，这完全是取绝对值函数造成的结果。从对应的相位特性上，看出它在该处突跳了一个 π，这也正是强制把幅特性反号的后果。这也解释了过去某些例题中画出的相特性中为什么常常出现幅度为 π 或 $-\pi$ 的跳跃。左边那个从 $-\pi$ 到 π 的幅度为 2π 的跳跃是另一回事，那是 MATLAB 用反正切函数 arctan 求复数幅角时只取 $-\pi\sim\pi$ 间的主值造成的。

当滤波器系数 $h(n)$ 很长时用 hc721 中的语句来计算符幅特性显然是很不方便的，有必要编写一个 MATLAB 子程序来实现符幅特性的计算。

7.2.3　计算符幅特性的 MATLAB 程序

线性相位滤波器有四种类型，如果编四个程序分别计算它们的符幅特性，那显然太麻烦，也没有充分利用计算机的起码智能。最好编一个通用的程序，它能自动判别滤波器的类型并能在(7.2.2 a)~(7.2.5 a)四个公式中，自动选择正确的符幅特性公式进行计算。把这个函数程序命名为 amplres.m，其实现方法当然不是惟一的，本书提供的程序如下。

```
function [A,w,type,tao]=amplres(h)
% 给定 FIR 滤波器系数求滤波器符幅特性
% h=FIR 滤波器的脉冲响应或分子系数向量
% A=滤波器的符幅特性
% w=取的频率向量，在 0 到 pi 之间分成 500 份，501 个点
% type=线性相位滤波器的类型
% tao=符幅特性的群迟延
%
N=length(h);tao=(N-1)/2;
L=floor((N-1)/2);                                   % 求滤波器阶次及符幅特性的阶次
n=1:L+1;w=[0:500]*pi/500;                           % 取滤波器频率向量
if all(abs(h(n) -h(N-n+1))<1e-10)                   % 判断滤波器系数若为对称
  A=2*h(n)*cos(((N+1)/2-n)'*w) - mod(N,2)*h(L+1);   % 对称条件下计算 A（两种类型）
  % 在 N=奇数时 h(L+1)多算一倍，要减掉。N 为偶数时，乘 mod(N,2)以取消这项。
  type =2-mod(N,2);                       % 判断并给出类型
```

```
    elseif all(abs(h(n)+h(N-n+1))<1e-10)&(h(L+1)*mod(N,2)==0)    % 系数若为反对称
    % N=奇时,h(L+1)为零是奇对称判别条件之一，N=偶时不要这条件，故乘以mod(N,2)。
A=2*h(n)*sin(((N+1)/2-n)'*w);                        % 反对称条件下计算A的公式（两种类型相同）
       type =4-mod(N,2);                              % 判断并给出类型
    else error('错误：这不是线性相位滤波器!')  % 滤波器系数非对称，报告错误
    end
```

在阅读这个程序时，注意以下几点：

（1）首先要注意这里的 n 比公式中的 n 大 1，这是因为 MATLAB 下标从 1 开始，而公式中的下标从零开始；公式中的下标 L 在程序中是 L+1，公式中的求和指标从 n=0 到 L 在程序中是 n=1 到 L+1，依次类推；

（2）关于判断序列对称和反对称的条件语句，似乎该用 all(h(n) -h(N-n+1))==0，程序中则采用了 all(abs(h(n)-h(N-n+1))<1e-10)。这是因为生成 h(n)和 h(N-n+1)中会有计算误差，两者往往无法完全相等，为了防止造成误判，要允许两者有一点差别；

（3）计算符幅特性时，用行向量 2h(n)与列向量 cos（(τ-n)ω）相乘，实现求和号下的相乘累加运算。计算类型 I 时，因为向量 2h(n)中的末一个元素为 2h(L+1)，比公式中的 h(L+1) 多了一倍，所以后面把它减去一个 h(L+1)；

（4）程序中采用了 mod(N,2)函数，来区分奇偶的不同效果，避免用多个条件转移语句。在 N 为奇数时它等于 1，而在 N 为偶数时它等于 0。用 mod(N,2)去乘一个变量，相当于在 N 为偶数时把这项变量取消。用了这个技巧，使适用于四种情况的计算程序程序变得如此简练。

例 7.2.2：给出由序列 h0=[3, -1,2, -3]为基础构成的四种类型的系数向量：

(1) h1=[3,-1,2,-3,5,-3,2,-1,3];　　　(2) h2= [3,-1,2,-3,-3,2,-1,3];

(3) h3=[3,-1,2,-3,0, 3,-2, 1,-3];　　(4) h4=[3, -1,2,-3,0,3,-2, 1,-3]

分别求出它们的符幅特性曲线，画在同一张图上做比较。

解：程序 hc722 如下：

```
h1=[3,-1,2,-3,5,-3,2,-1,3];
[A1,w1,typea,tao1]=amplres(h1);typea
subplot(2,2,1),plot(w1,A1),grid on
……
```

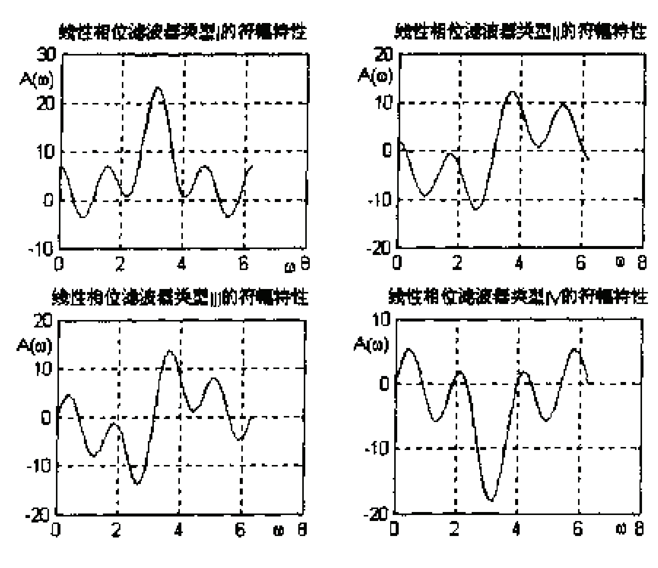


图 7.2.2　四种类型线性相位滤波器的符幅特性

（取 0～2π 的全频段）

对 h2,h3,h4 都进行同样的处理，把结果分别画在其他三个子图上。另外，把 amplres 子程序中的 w 向量乘以 2，以便把频率区间扩大到 2π，观察它们的对称性（将来是用不到的，所以子程序中不需要将 w 扩展）。可依次得到这四个滤波器的系数向量的类型为 typea=1,typeb=2,typec=3 和 typed=4，程序运行结果见图 7.2.2。这个图中的符幅特性曲线可以进一步验证表 7.2.1 中关于以下四种类型滤波器符幅特性的特征。

类型 I：幅特性关于 $\omega=\pi$ 对称，在 $\omega=0$ 和 $\omega=\pi$ 处可以取任何值；

类型 II：幅特性关于 $\omega=\pi$ 反对称，在

ω=0 可以取任何值，在 $\omega=\pi$ 处必定为零；

类型 III：幅特性关于 $\omega=\pi$ 反对称，在 ω=0 和 $\omega=\pi$ 处都必定为零；

类型 IV：幅特性关于 $\omega=\pi$ 对称，在 ω=0 处必定为零，而 $\omega=\pi$ 处可以取任何值。

注意这个结果与“实序列的幅特性偶对称”的结论并不矛盾。如果把符幅特性的负的部分翻到横坐标上方，则四种情况的幅特性对 ω=0 都是偶对称的。

7.2.4　线性相位滤波器的零极点分布

FIR 滤波器的零点是它的系数多项式的根，可由 roots(*b*)求得。要注意虽然分母没有多项式，只是一个常数，但并非没有极点。因为零极点应该从正幂传递函数多项式来判断。

$$b_0+b_1z^{-1}+\cdots+b_M\ z^{-M}=\frac{b_0z^M+b_1z^{M-1}+\cdots+b_M}{z^M} \tag{7.2.13}$$

所以它具有与零点数目相同的 M 个极点，它们集中在 z 平面的原点处，成为 M 重极点。它离开单位圆的圆周最远，因此不存在稳定性问题。

对于类型 I 和类型 II 的线性相位滤波器，它的系数满足对称条件：

$$b_0=b_M,b_1=b_{M-1},\ldots,b_n=b_{M-n}\qquad (n=0,1,M/2) \tag{7.2.14}$$

可以证明，对于具有此种对称性的系数向量，如果 z_1 是(7.2.13)式的分子多项式的根，那么 $q_1=1/z_1$ 也是这个多项式的根。事实上，若

$$b_0z_1^M+b_1z_1^{M-1}+\cdots+b_M=0 \tag{7.2.15}$$

将 $q_1=1/z_1=z_1^{-1}$ 代入此分子多项式(7.2.15)中的 z_1 可得

$$b_0z_1^{-M}+b_1z_1^{-M+1}+\cdots+b_{M-1}z_1^{-1}+b_M=z_1^{-M}(b_0+b_1z_1+\cdots+b_{M-1}z_1^{M-1}+b_Mz_1^M)$$

利用(7.2.14)式将此式括号中的系数对换：

$$b_0+b_1z_1+\cdots+b_{M-1}z_1^{M-1}+b_Mz_1^M=b_M+b_{M-1}z_1+\cdots+b_1z_1^{M-1}+b_0z_1^M$$

显然，上式的右端就是(7.2.15)式的左端，当然它等于零。这就表明 $q_1=1/z_1$＝也是(7.2.15)的根，所以它也是系统的零点。

如果 z_1 是复数零点，那么根据实系数多项式的性质，它的共轭数 z_1*也是复数零点，现在又证明了它的倒数 $1/z_1$ 也是零点，因而倒数的共轭数 z_1*也是复数零点。四个复数零点必成组出现，如图 7.2.3 所示。这是类型 I 和类型 II 的线性相位滤波器的一个特点。如果是实数零点，那就只有 z_1 和它的倒数 $1/z_1$ 两个零点成对出现。

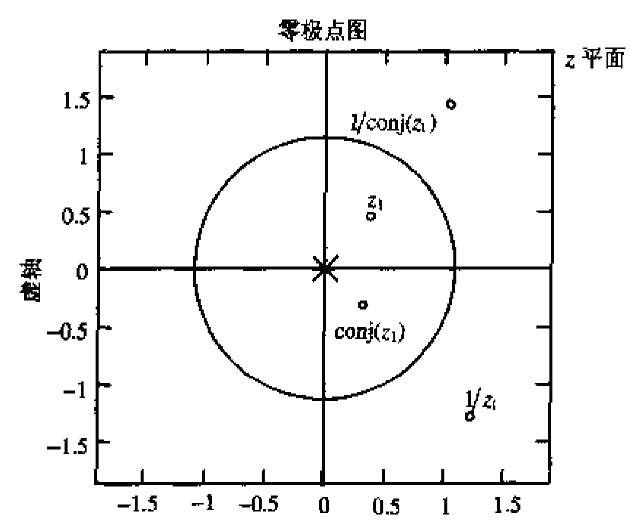


图 7.2.3　线性相位滤波器的极点和成组的四个复零点

7.3　设计 FIR 滤波器的窗函数法

7.3.1　对脉冲响应进行截断的分析方法

设计滤波器的基本思想是让待设计的实际滤波器逼近理想特性。在第 5.5.1 节中曾经研究了理想低通滤波器的频率特性应为：振幅（符幅）特性在通带内为 1，阻带内为 0；在通带内的相位特性与 ω 成线性关系。即（注：$\tau=n_0$）

$$H_{\rm d}({\rm e}^{{\rm j}\omega})=\begin{cases}1\times{\rm e}^{-{\rm j}\omega n_0} & |\omega|\leqslant\omega_c\\ 0 & \omega_c<\omega\leqslant\pi\end{cases} \tag{5.5.1}$$

它对应的理想单位脉冲响应 $h_{\rm d}(n)$ 为

$$h_{\rm d}(n)=\frac{1}{2\pi}\int_{-\omega_c}^{\omega_c}{\rm e}^{-{\rm j}n_0\omega}{\rm e}^{{\rm j}n\omega}{\rm d}\omega=\frac{\sin(\omega_{\rm c}(n-n_0))}{\pi(n-n_0)} \tag{5.5.4}$$

可见，理想低通滤波器的单位脉冲响应 $h_{\rm d}(n)$ 是无限长的非因果序列。为了构造物理上可实现的长度为 N 的因果线性相位滤波器，必须将 $h_{\rm d}(n)$ 截取长度为 N 的一段，由于起点位置取为 n=0 处，它的中心点位置应取为 $n_0=(N-1)/2$，从而保证截取的那段序列对$(N-1)/2$ 对称。设截取的脉冲响应用 $h(n)$表示，把它作为实际滤波器的系数向量。从第 2 章知道，截取相当于和一个矩形窗相乘，即

$$h(n)=h_{\rm d}(n)R_N(n) \tag{7.3.1}$$

式中 $R_N(n)$ 是一个长度为 N 的矩形序列，其中心位置对准 $h_{\rm d}(n)$ 的中心点，如图 5.5.3 下图所示，截取的一段 $h(n)$关于$(N-1)/2$ 点对称，保证所设计的滤波器具有线性相位。因此，实际滤波器的系统函数 $H(z)$为 $H(z)=\sum\limits_{n=0}^{N-1}h(n)z^{-n}$ 用一个有限长的序列 $h(n)$去代替 $h_{\rm d}(n)$，肯定会引起误差，表现在频域就是通常所说的吉布斯（Gibbs）效应。该效应引起通带和阻带内的波动性，吉布斯效应是由于将 $h_{\rm d}(n)$ 直接截断引起的，因此，也称为截断效应。用图 7.3.1 可以说明截断对实际滤波器频率特性的影响是如何产生的。

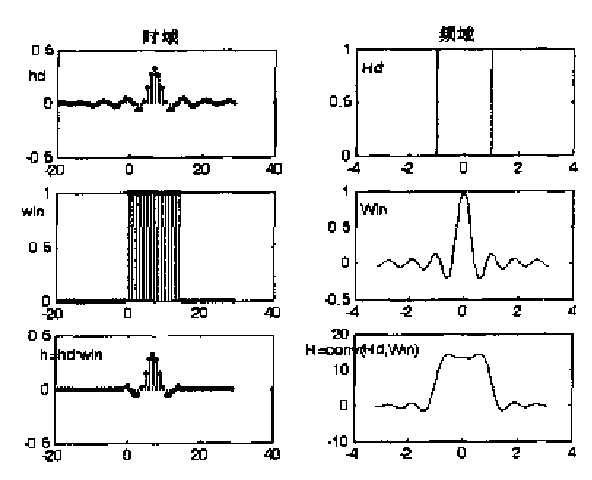


图 7.3.1　理想低通滤波器的单位脉冲响应及矩形窗频域卷积

在图 7.3.1 中共有六个子图，左边的一列是时域的序列，右边一列则是左边序列所对应的频谱图或频率特性。第一行是理想脉冲响应，左边的时域波形应向左右无限延伸；而其右边的频率特性就是带宽有限的低通滤波器理想特性；第二行是矩形窗序列，是用来截取理想脉冲响应的，它的时域波形具有有限长度，而它的频谱特性却向左右无限延伸。左边第三行的序列是上两行序列相乘的结果，也就是截取的实际滤波器的系数向量，根据傅里叶变换的性质——时域的相乘对应于频域的卷积，右列第三行子图的频率特性应该是右列上面两个子图波形卷积的结果。它表明了脉冲响应的截断使实际频率特性与理想特性产

生的误差或畸变，在滤波器设计中所关心的正是这个问题。

下面将定量地分析这种截断效应，讨论如何选用适当的窗函数减少截断效应，设计满足特定技术要求的 FIR 线性相位滤波器。

7.3.2　矩形窗截断的定量分析

上节介绍了用窗函数法设计 FIR 滤波器的思路。因为 $H_d(\omega)=\sum_{n=-\infty}^{\infty}h_d(n)e^{-j\omega n}$ 可以看成把频谱函数展开为傅里叶级数，其各个分量的系数为 $h_d(n)$，也就是对应的单位脉冲序列。设计 FIR 滤波器的任务就是，找到有限个傅里叶级数系数 $h(n)$，以有限项傅里叶级数去近似代替无限项傅里叶级数。这样在一些频率不连续点附近会引起较大误差。这种误差的效果就是前面说的截断效应。为了减少这一效应，要掌握这样一些规则。首先，选取傅里叶级数的项数越多，引起的误差就越小，但 $h(n)$长度增加也使运算速度降低和成本加大，应在满足技术要求的条件下，尽量减少 $h(n)$的长度；另一个思路是，不应该只是简单地截断，可以对取出的系数进行加权修改，就是用适当的窗函数形状来加权。因此，从这一角度来说，窗函数法也称为傅里叶级数法。

对（7.3.1）式进行傅里叶变换，根据复卷积定理得到：

$$H(e^{j\omega})=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(e^{j\omega})A_{RN}(e^{j(\omega-\theta)})d\theta \tag{7.3.2}$$

式中，$H_d(e^{j\theta})$和$A_{RN}(e^{j\omega})$ 分别是 $h_d(n)$和 $R_N(n)$的傅里叶变换。卷积要涉及被卷积函数的正负号。预期幅特性$|H_d(\omega)|$常为正，而 $R_N(n)$的傅里叶变换要用其符幅特性表示如下：

$$\text{DTFT}[R_N(n)]=\sum_{n=0}^{N-1}R_N(n)e^{-j\omega n}=e^{-j(N-1)\omega/2}\frac{\sin(\omega N/2)}{\sin(\omega/2)}=A_{RN}(\omega)e^{-j\tau\omega}$$

式中

$$A_{RN}(\omega)=\frac{\sin(\omega N/2)}{\sin(\omega/2)},\qquad \tau=\frac{N-1}{2} \tag{7.3.3}$$

把矩形窗的频率特性分解出线性相位部分$-\tau\omega$ 后，$A_{RN}(\omega)$部分就是符幅特性函数；在例 3.2.1 中已经求出它的波形特征，中间正值的部分为主瓣，它最宽，占了 $\frac{4\pi}{NT}$ ，其他为副瓣，共有 N-2 个，每个宽度为 $\frac{2\pi}{NT}$ 。再把理想滤波器频率特性也写成线性相位形式：

$$H_d(e^{j\theta})=H_d(\omega)\cdot e^{-j\theta(\omega)}=1\cdot e^{-j\tau\omega} \tag{7.3.4}$$

将(7.3.3)和(7.3.4)代入(7.3.2)，得到

$$\begin{aligned}H(e^{j\omega})&=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(\theta)e^{-j\theta\tau}A_{RN}(\omega-\theta)e^{-j(\omega-\theta)}d\theta\\&=e^{-j\omega L}\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(\theta)A_{RN}(\omega-\theta)d\theta\end{aligned} \tag{7.3.5}$$

（7.3.5）式也把线性相位部分和符幅特性部分分开了，所以实际滤波器的符幅特性为理想滤波器符幅特性与矩形窗函数符幅特性的卷积，写成

$$H(\omega)=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(\theta)A_{RN}(\omega-\theta)d\theta \tag{7.3.6}$$

矩形窗与理想低通滤波器符幅特性的频域卷积过程见图 7.3.2。图 7.3.2（f）表示 $H_d(\omega)$

与 $A_{RN}(\omega)$卷积形成的 $H(\omega)$波形。当 $\omega=0$ 时，$H(0)$等于图 7.3.2（a）与（b）两波形乘积的积分，实际上只要取$-\omega_c\sim\omega_c$的一段波形的积分，当 $\omega_c \gg 2\pi/N$ 时，近似等于$\pm\pi$ 之间波形的积分。将 $H(0)$值归一化到 1，当 $\omega=\omega_c$ 时，情况如图 7.3.2（c）所示，当 $\omega \gg 2\pi/N$ 时，积分近似为 $A_{RN}(\omega)$ 一半波形的积分，其值为 1/2。当 $\omega=\omega_c+2\pi/N$ 时，如图 7.3.2（e）所示。

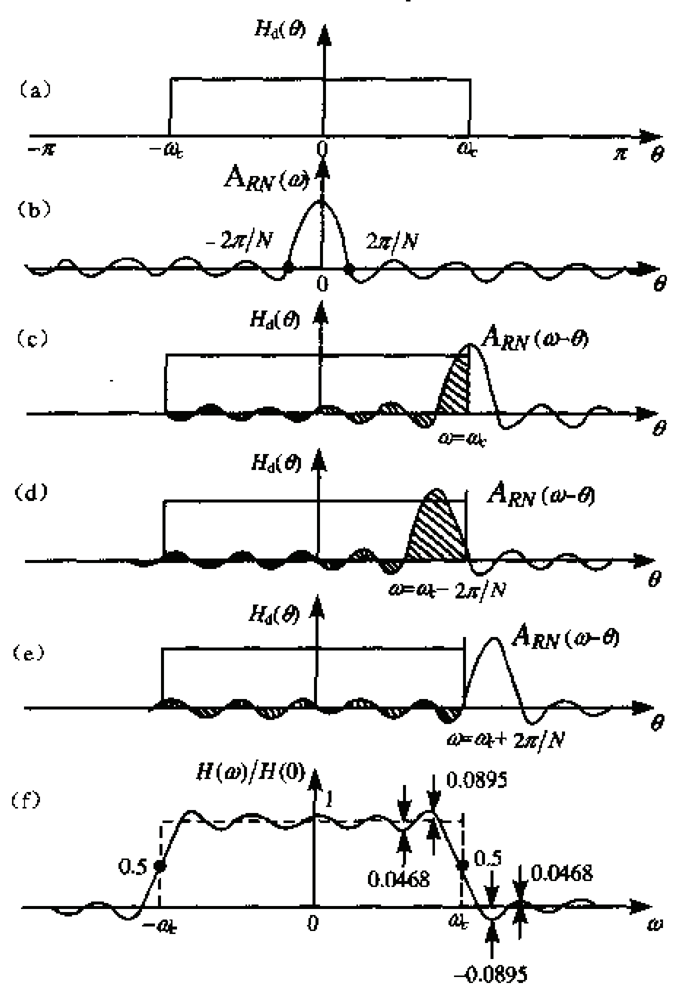


图 7.3.2　矩形窗与理想低通滤波器符幅特性的频域卷积

$A_{RN}(\omega)$主瓣完全移到积分区间外边，因为最大的一个负峰完全在区间$[-\omega_c,\omega_c]$中，因此 $H(\omega)$在该点形成最大的负峰。相应地，当 $\omega=\omega_c-2\pi/N$ 时，如图 7.3.2（d）所示。$A_{RN}(\omega)$主瓣完全在区间$\pm\omega_c$之间，而最大的一个负峰移到区间$[-\omega_c, \omega_c]$之外，因此，$H(\omega)$最大的正峰与最大的负峰对应的频率相距 $4\pi/N$。通过以上分析可知，对 $h_d(n)$加矩形窗处理后，H（ω）和原理想低通滤波器 $H_d(\omega)$的差别有以下两点：

（1）在理想特性不连续点 $\omega=\omega_c$ 附近形成过渡带。过渡带的宽度，近似等于 $R_N(\omega)$主瓣宽度，即 $4\pi/N$。

（2）通带内增加了波动，最大的峰值在 $\omega=\omega_c-2\pi/N$ 处。阻带内产生了余振，最大的负峰在 $\omega=\omega_c+2\pi/N$ 处。通带与阻带中波动的情况与窗函数的符幅谱有关。$A_{RN}(\omega)$波动愈密（N 加大时），通带、阻带内波动也愈密，$A_{RN}(\omega)$旁瓣的大小直接影响 $H(\omega)$波动的大小。

以上两点就是对 $h_d(n)$用矩形窗截断后，其效应在频域的反映，称为吉布斯效应。这种效应直接影响滤波器的性能。通带内的波动影响滤波器通带中的信号通过的均匀性；阻带内的波动使某些频率成分在阻带内不能充分抑制，可能使最小衰减不满足技术要求；同时，在通

带与阻带连接处，又产生了一个过渡带，一般滤波器都要求过渡带愈窄愈好。

直观上，增加矩形窗口的宽度，即加大 N，似乎可以减少吉布斯效应的影响。下面分析一下 N 加大时 $A_{RN}(\omega)$的变化，主要看在主瓣附近的情况。在主瓣附近，$x \ll 1$（比如 0.3），近似有

$$A_{RN}(\omega) \approx \frac{\sin(\omega N/2)}{\omega/2} = \frac{\sin Nx}{x} \tag{7.3.7}$$

在 x=0 的主瓣中心，$\frac{\sin Nx}{x} = N$。在 Nx=0~π 区间，$\sin Nx$ 常为正，所以这是主瓣区。在 $x=\pi/N$ 时，出现第一个零点。而在 $x=3\pi/2N$ 处，$\sin Nx$ 出现第一个负峰，其幅度为 $R(\frac{1.5\pi}{N}) = \frac{\sin 1.5\pi}{1.5\pi/N} = -0.2122N$。随着 N 的加大，该函数的数值主瓣幅度加高，同时旁瓣也加高，但主瓣和旁瓣幅度相对值保持不变；另一方面，N 的加大引起波动的频度加大。这个近似式只适用于 $x \ll 1$ 的情况，所以，在负峰的地方已不准确，不能按此推论了。

我们真正关心的不是 $A_{RN}(\omega)$的波动，而是 $H(\omega)$在突变点 $\omega=\omega_c$ 附近的波动，$H(\omega)$是 $A_{RN}(\omega)$与理想幅特性卷积的结果，$A_{RN}(\omega)$从$-\infty$逐步右移，进入积分区间$-\omega_c \sim \omega_c$，形成 $H(\omega)$ 的过程实际上就是对 $A_{RN}(\omega)$进行积分，积分的结果称为累积符幅函数。因为用解析方法不能得到闭合的结果，这里就用 MATLAB 进行数值积分。程序 hc731 如下：

```
N=10;dx=0.01;                                   % 设置初值
x=[-pi/2:dx:pi/2]+1e-10;                        % 设置自变量数组，偏置微量
AR=sin(N*x)./sin(x);                            % 求矩形窗符幅函数值，不能用近似公式
H=cumtrapz(R)*dx/pi;                            % 求累积符幅函数，cumtrapz 为数字积分函数
subplot(2,1,1),plot(x,AR),grid on               % 画符幅函数曲线
subplot(2,1,2),plot(x,AR),grid on               % 画累积符幅函数曲线
delta1=max(H) -1,                               % 求通带最大波动（绝对值）
delta2=abs(min(H)),                             % 求阻带最大波动（绝对值）
Rp=-20*log10((1-delta1)./(1+delta1))            %    由 delta1 及 delta2 求相对值 Rp
As=-20*log10(delta2./(1+delta1))                %    由 delta1 及 delta2 求相对值 As
```

在编这个程序时要考虑以下几点：

（1）不能也不必取符幅函数的近似（7.3.7）式，必须用（7.3.3）式；

（2）（7.3.3）式表示的是一个周期函数，其周期为 π，因此自变量 x 的选取范围为 [$-\pi/2$，$\pi/2$]；

（3）由于在某些点可能出现 0/0 的情况，有意把自变量偏移一个微量 1^−10(在 MATLAB 中写成 1e−10,因为有时嫌取 eps 太小)；

（4）求累积符幅函数时，采用了数值积分函数 cumtrapz,为了把它的积分终值归一化到一，一是要乘以积分步长 dx，二是要除以积分区间的长度 π。

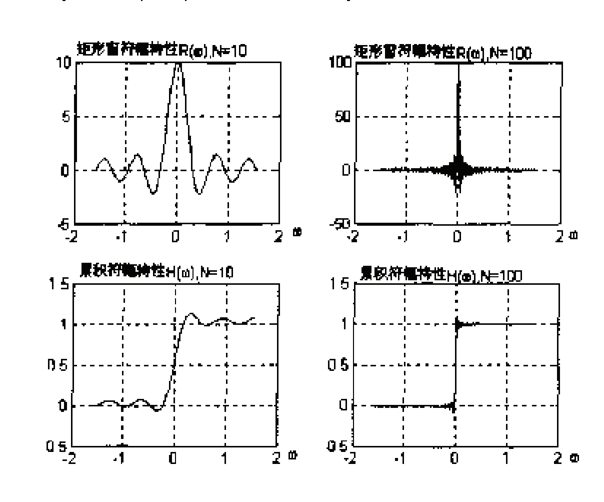


图 7.3.3　N=10（左）和 N=100（右）时矩形窗的符幅特性和累积符幅特性

用这个程序，按 N=10 和 100 两种情况进行计算，得到的波形见图 7.3.2 和图 7.3.3。从图中可以看到，增加 N 不能减少通带波动和阻带波动的值，它只能减小过渡带的宽度。虽然

大多数滤波器都希望减小过渡带，但必须考虑付出的代价是运算次数大大增加。两种情况算出的阻带波动分别为 0.06 和 0.08 左右，说明 N 对阻带波动幅度的影响很小。一般书上提供的矩形窗的累积符幅特性负峰值为 0.089，相当于-21dB，近似与 N 无关。

近似估计幅特性过渡带宽度的方法是从 H 曲线的负峰到正峰所占频带，也就是 $R_N(\omega)$主瓣的宽度 $4\pi/N$。不过这种估计太粗了，它把波形上两个峰值附近的平坦部分也当成过渡带是不对的。可以用程序计算符幅特性上升边的宽度，按 H 从 0.1 $H(0)$增加到 0.9$H(0)$所占的带宽，那样算出的结果，过渡带带宽为 $1.8\pi/N$，只有近似值的不到一半。

7.3.3 几种其他窗函数的介绍

归纳上节，调整窗口长度 N 可以有效地控制过渡带的宽度，但对减少带内波动以及加大阻带的衰减不起作用。许多科学家就从窗函数的形状上寻找解决方法。他们力求找到适当的窗函数形状，使其符幅函数的主瓣包含更多的能量，以减小其旁瓣幅度。旁瓣的减小可使通带、阻带波动减小，从而加大阻带衰减。但总是以加宽过渡带为代价的。下面介绍几种常用的窗函数 $wd_x(n)$，这些窗函数大都是以提出者的名字命名的，将其缩写字母作为下标 x。实际滤波器的系数向量 $h(n)$就由理想滤波器脉冲响应 $h_d(n)$与窗函数 $wd_x(n)$相乘得到：

$$h(n)=h_d(n)\cdot wd_x(n)$$

下面列举一些常用的窗函数：

（1）矩形窗（Rectangular Window）

$$wd_R(n)=R_N(n) \tag{7.3.8}$$

前面已分析过，按照（7.3.1）式，其用符幅特性表示的频率特性为

$$wd_R(e^{j\omega})=\frac{\sin(\omega N/2)}{\sin(\omega/2)}e^{-j(N-1)\omega/2}=A_{RN}(\omega)e^{-j\tau\omega} \tag{7.3.8a}$$

它的主瓣宽度为 $4\pi/N$，第一副瓣比主瓣低 13 dB。

（2）三角形窗（Bartlett　Window）

$$wd_{Br}(n)=\begin{cases}2n/(n-1) & 0\leqslant n\leqslant (N-1)/2\\ 2-2n/(n-1) & (N-1)/2<n\leqslant N-1\end{cases} \tag{7.3.9}$$

其用符幅特性表示的频率特性为

$$wd_{Br}(e^{j\omega})=\frac{2}{N}\left[\frac{\sin(\omega N/4)}{\sin(\omega/2)}\right]^2 e^{-j[\omega+(N-1)\omega/2]} \tag{7.3.10}$$

其主瓣宽度为 $8\pi/N$，第一副瓣比主瓣低 26dB。

（3）汉宁（Hanning）窗——升余弦窗

$$wd_{Hn}(n)=0.5\left[1-\cos\left(\frac{2\pi n}{N-1}\right)\right]R_N(n) \tag{7.3.11}$$

由于有函数 amplres 可以方便地从窗函数 $wd(n)$直接计算符幅函数 $A(\omega)$的数值，以下本书将不再列出其他窗函数符幅函数的冗繁的闭合形式。计算说明，汉宁窗的符幅函数的能量更集中在主瓣中。但代价是主瓣宽度加宽到 $8\pi/N$，因而过渡带宽度也相应增加。

（4）哈明（Hamming）窗——改进的升弦窗

$$wd_{Hn}(n)=\left[0.54-0.46\cos\left(\frac{2\pi n}{N-1}\right)\right]R_N(n) \tag{7.3.12}$$

这种改进的升余弦窗，主瓣集中的能量约占 99．96%，第一旁瓣的峰值比主瓣小 40 dB，但主瓣宽度和汉宁窗相同，仍为 $8\pi/N$。

（5）布莱克曼（Blackman）窗

$$wd_{\mathrm{Bl}}(n)=\left[0.42-0.5\cos\left(\frac{2\pi n}{N-1}\right)+0.08\cos\left(\frac{4\pi n}{N-1}\right)\right]R_N(n) \tag{7.3.13}$$

（6）凯泽-贝塞尔窗（Kaiaser-Bassel Window）

这是最有用且最优的窗结构之一，对于给定的阻带衰减，它提供了最小的主瓣宽度，也就是最陡的过渡带。就这点而言，它是最优的。窗函数由凯泽(J. F. Kaiser)提出，由下式给出：

$$wd_{\mathrm{K}}(n)=\frac{I_0\left[\beta\sqrt{1-\left(1-\dfrac{2n}{N-1}\right)^2}\right]}{I_0(\beta)}\qquad 0\leqslant n\leqslant N-1 \tag{7.3.14}$$

式中 $I_0[*]$是零阶第一类修正贝塞尔函数，β 为依赖于 N 的参数，选择 N 可产生各种过渡带宽和接近最优的阻带衰减。对于相同的 N，此窗可以靠选择 β 提供不同的过渡带宽，这是其他窗做不到的。凯泽还给出了经验公式，在给定 ω_p,ω_s,R_p 和 A_s 时，可以用以下的式子计算滤波器长度 N 和参数 β:

$$N\approx\left\lceil\frac{A_s-7.95}{14.36\Delta f}\right\rceil+2 \tag{7.3.15}$$

其中 $\Delta f=(\omega_s-\omega_p)/2\pi$　为归一化过渡带宽。

$$\beta=\begin{cases}0.1102(A_s-8.7) & A_s>50\\ 0.5842(A_s-21)^{0.4}+0.07886(A_s-21) & 21\leqslant A_s\leqslant 50\\ 0 & A_s<21\end{cases} \tag{7.3.16}$$

这种窗函数取不同 β 值的性能归纳在表 7.3.1 中，六种窗函数的基本参数归纳在表 7.3.2 中，可供设计时参考。

7.3.4　MATLAB 提供的窗函数

MATLAB 提供了相应的子程序来实现本节中的窗函数，对应于前面举出的六种，分别为：

```
wd=boxcar(N)          % 数组wd中返回N点矩形窗函数
wd=triang(N)          % 数组wd中返回N点Bartlett（三角）窗函数
wd=hanning(N)         % 数组wd中返回N点汉宁窗函数
wd=hamming(N)         % 数组wd中返回N点哈明窗函数
wd=blackman(N)        % 数组wd中返回N点布莱克曼函数
wd=kaiser(N,beta)     % 数组wd中返回给定beta值时N点凯泽窗函数
```

其他的还有七八种，不一一列举。

这些函数的输入变元一般只要窗函数的长度 N 就够了，只有凯泽窗还需要规定 β 值。输出变元就是中心值归一化为 1 的窗函数序列 wd，它是列向量。利用语句 stem([1:N]',wd)就可以画出窗函数序列的形状，例如执行具有下列语句的程序 hc732 就可以得到综合表示几种窗函数形状的图 7.3.4。MATLAB 是用颜色区分并标注各条曲线的，在黑白印刷时无法表示，所以这里就只好使用手工画的图了。读者自己在计算机上可以得到相似的、以不同颜色区分的图形。

```
N=45; n=1:N; beta=5.658;
wdrc=boxcar(N); wdbt=triang(N); wdhn=hanning(N);
wdhm=hamming(N); wdbl=blackman(N); wdks=kaiser(N,beta);
plot(n',[wdrc,wdbt,wdhn,wdhm,wdbl,wdks])
legend('rc','bt','hn','hm','bl','ks') % 线型标注
```

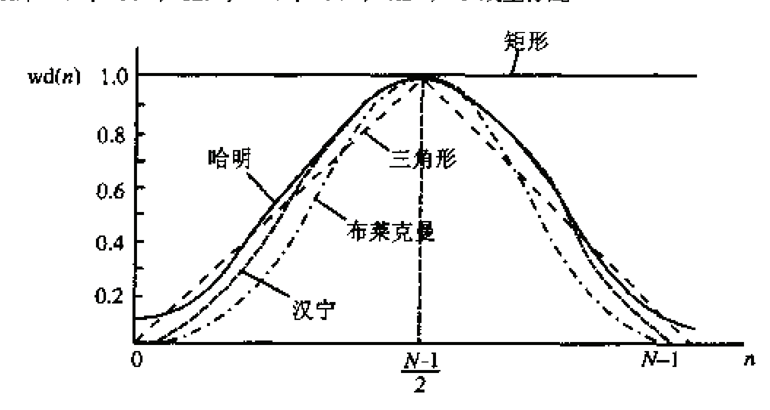


图 7.3.4　常用的几种窗函数形状

除了矩形窗具有正负变化的符幅特性以外，其他各种窗函数都没有负的幅特性，所以其幅特性就等于符幅特性，计算时可以调用信号处理工具箱中的 freqz 函数，进而通过求对数得到以 dB 为单位的幅特性。

例 7.3.3：求出长度为 51 的布莱克曼窗的符幅特性，累积符幅特性，以及以分贝表示的幅特性和累积符幅特性曲线，用四个子图画在同一张图上。

解：解此题的 MATLAB 程序核心语句如下：

```
N=51; n=1:N; wd=triang(N);                 % 给出三角窗参数，产生窗函数
[A,omega]=amplres(wd');                    % 调用 amplres，求符幅函数，变 wd 为行向量
Ad=[fliplr(A),A(2:end)];                   % 将符幅函数对称扩展至负频率区
omegad=[-fliplr(omega),omega(2:end)];      % 将频率对称扩展至负频率区
Gd=cumtrapz(Ad)*(omega(2)-omega(1))/pi;    % 积分求累积符幅函数 Gd
AdB=20*log10(abs(Ad));                     % 将符幅函数转换为分贝
GdB=20*log10(Gd);                          % 将累积符幅函数转换为分贝
```

以下为图形分割及绘图语句 plot(omegad,Ad)等，分别画出 Ad,Gd,AdB,GdB 四个变量随 omegad 的变化曲线如图 7.3.5 所示。注意程序中最后两行求 AdB，GdB 的语句是有悬念的，因为负数没有对数，故符幅特性中负的值不能用 dB 表示。如果出现这种情况，程序就会停止运行并给出出错告警。这两条语句实际上成了检验语句，在三角窗条件运行本程序，不出现停机，说明其符幅特性全为正数，因此它和幅特性相同。

从曲线和详细分析结果数据可以得知：

（1）三角窗的符幅特性 A 已经没有负值的部分了，所以其符幅特性与幅特性相同。为了减少滤波器的阻带和通带波动，所有的窗函数都避免符幅特性中出现负值；

（2）键入 whos 命令，可知 Ad,Gd,AdB 和 GdB 的长度都是 1001；

（3）键入 Ad(501)，得知其中点的峰值为 26，键入 max(Ad(1:450))得知其第一个副瓣的幅度为 1.2392，由 20*log10(26/1.2392)得到其第一副瓣比最大值低 26.4dB。比矩形窗的 13dB 强多了。

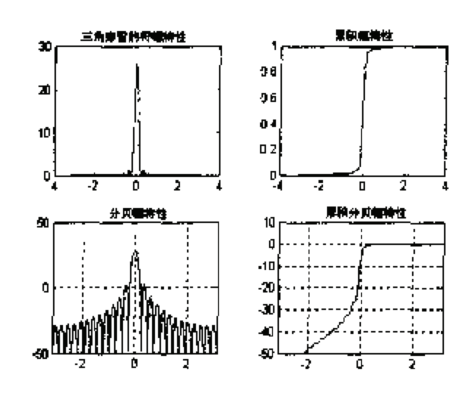


图7.3.5　三解窗函数的符幅特性和累积符幅特性(上图为数值，下图为分贝)

窗函数的特性早有前人给做详细计算并列成表7.3.1和7.3.2。这个例子说明如何利用现在的计算机软硬件资源，快速地求出这些表格中的内容。这是当代人的条件优于前人的地方，应该充分利用和发挥。

表7.3.1　六种窗函数的特性表

窗函数	旁瓣峰值 / dB	近似过渡带宽	精确过渡带宽	阻带最小衰减 / dB
矩形窗	-13	$4\pi/N$	$1.8\pi/N$	21
三角形窗	-25	$8\pi/N$	$6.1\pi/N$	25
汉宁窗	-31	$8\pi/N$	$6.2\pi/N$	44
哈明窗	-41	$8\pi/N$	$6.6\pi/N$	53
布莱克曼窗	-57	$12\pi/N$	$11\pi/N$	74
凯泽窗(β=7.865)	-57		$10\pi/N$	80

表7.3.2　凯泽窗参数对滤波器性能的影响

β	过渡带宽（π/N）	通带波纹（dB）	阻带最小衰减（dB）
2.120	3.00	±0.27	30
3.384	4.46	±0.0864	40
4.538	5.86	±0.0274	50
5.568	7.24	±0.008 68	60
6.764	8.64	±0.002 75	70
7.865	10.0	±0.000 868	80
8.960	11.4	±0.000 275	90
10.056	10.8	±0.000 087	100

7.3.5　用窗函数设计FIR滤波器的步骤

用窗函数设计FIR滤波器的步骤如下：

(1) 根据对过渡带宽及阻带衰减要求，选择窗函数的类型并估计窗口长度 N (或阶数 $M=N-1$)。窗函数类型可根据其阻带最小衰减（见表7.3.1）A_s 的条件独立选择，因为其长度 N 对阻带最小衰减没有影响。在确定窗函数类型后，就可根据过渡带宽小于给定指标的条件，确定所拟用的窗函数长度 N。设待求滤波器的过渡带为$\Delta\omega$，它近似与窗口长度 N 成反比（见表7.3.1）。窗函数类型确定后，其计算公式也确定了。不过这些公式是近似的，得出的长度还要在计算中逐步修正。原则是在保证阻带衰减满足要求的情况下，尽量选择较小的 N。在

N 和窗函数类型确定后，即可调用 MATLAB 中的窗函数(见第 7.3.3 节)求出 wd。

(2) 根据待求滤波器的理想频率响应求出单位脉冲响应 $h_d(n)$。如果给出待求滤波器的频率响应为 $H_d(e^{j\omega})$，那么单位脉冲响应用下面的傅里叶反变换式求出：

$$h_d(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(e^{j\omega})e^{j\omega n}d\omega \tag{7.3.17}$$

在一般情况下，$h_d(n)$是不能用封闭公式表示的，需要采用数值方法。从 $\omega=0$ 到 $\omega=2\pi$ 采样 N 点，用第 4 章中的方法靠 IDFT 来求出。对于分段为常数的理想幅频特性，可以利用 MATLAB 根据(5.5.4)式编出一个计算理想低通滤波器频率特性的子程序。其他选频滤波器特性都可以由低通特性合成，因此可以调用这个子程序进行代数运算得到其理想特性。此子程序如下：

```
function hd = ideallp(wc,N);
% hd = 点 0 到 N-1 之间的理想脉冲响应
% wc = 截止频率(弧度)
% N = 理想滤波器的长度
tao = (N-1)/2;
n = [0: (N-1)];
m = n - tao + eps; % 加一个小数以避免零作除数
hd = sin(wc*m) ./ (pi*m);
```

这个算式程序实际上在 (5.5.4) 式中已经得到，这里只是使它通用化。对于高通、带通、带阻以及多段滤波特性而言，只要其理想幅特性是矩形的，都可以表为几个理想低通滤波器的叠加。比如 ω_{c1}~ ω_{c2} 的理想带通滤波特性可以用通带为 ω_{c2} 的理想低通滤波特性减去通带为 ω_{c1} 的理想低通滤波特性而得到，如图 7.3.6 所示。

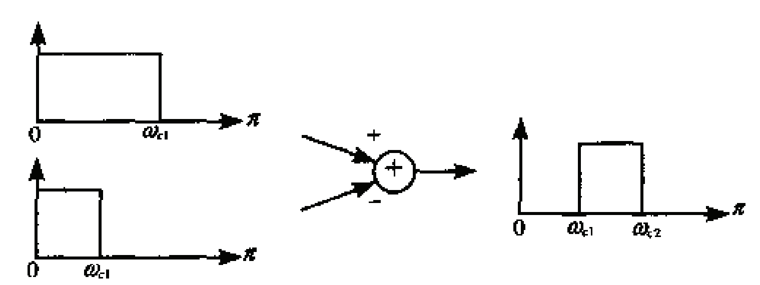


图 7.3.6　由两个低通滤波器得到的理想带通滤波器

而对应于它的理想脉冲响应也可根据线性叠加的原理，在给定 N 时，由 MATLAB 语句

```
hd= ideallp(wc2,N)- ideallp(wc1,N)
```

求得。对于其他矩形滤波特性，读者可以自行推理，得知其理想脉冲响应的求法。

(3) 计算滤波器的单位脉冲响应 $h(n)$。它是理想脉冲响应和窗函数的乘积 $h(n)=h_d(n)*wd(n)$，在 MATLAB 中用点乘命令表示为 $h=hd.*wd$。

h_d 在步骤(2)中求得，它是一个行向量；wd 在步骤(1)中求得，它也必须是一个同长度的行向量。但 MATLAB 中调用的窗函数产生的常常是列向量，这时就需要把它转置一次。窗函数 wd 对 $(N-1)/2$ 点是偶对称的，因此产生的 $h(n)$的对称性完全取决于理想脉冲响应 $h_d(n)$ 的对称性，与窗函数无关。

(4) 验算技术指标是否满足要求。为了计算数字滤波器在频域中的特性，可以用 freqz 子程序。为了方便检查有关滤波器设计的所有指标，利用这个子程序，我们开发了一个扩展版

本，叫做 myfreqz，它同时给出绝对的和相对的 dB 值幅度响应，相位响应，以及群延迟响应。

```
function [db,mag,pha,grd,w] = myfreqz(b,a);
% freqz 子程序的改进版本
% ----------------------------------
% [db,mag,pha,grd,w] =myfreqz (b,a);
% db = [0 到pi 弧度]区间内的相对振幅(db)
% mag = [0 到pi 弧度]区间内的绝对振幅
% pha = [0 到pi 弧度]区间内的相位响应
% grd = [0 到pi 弧度]区间内的群延迟
% w =  [0 到pi 弧度]区间内的 501 个频率样本向量
% b = Ha(z)的分子多项式系数(对 FIR b=h)
% a = Ha(z)的分母多项式系数(对 FIR: a=[1])
[H,w] = freqz(b,a,1000,'whole');
   H = (H(1:1:501))'; w = (w(1:1:501))';
 mag = abs(H);                              % 计算绝对振幅
   db = 20*log10((mag+eps)/max(mag)); % 计算相对振幅(db)
 pha = angle(H);                            % 计算相位响应
 grd = grpdelay(b,a,w);                     % 计算群延迟
```

将设计出的滤波器系数向量代入，调用这个子程序，就可以得到它的频率响应的所有指标。

(5)如果不满足要求，可根据具体情况，调整窗函数类型或长度，重复(1)、(2)、(3)、(4)步，直到满足要求为止。

例 7.3.4：用矩形窗、汉宁窗和布莱克曼窗设计 FIR 低通滤波器，设 N=11，ω_c=0.2π。

解：因为题中已给出了窗函数类型及其长度，解此题时可以省去步骤(1)，直接从步骤(2)开始。由此编出的 MATLAB 程序 hc734 如下：

```
wc=0.2*pi; N=11;
hd= ideallp(wc,N)                          % 用wc=0.2*pi 的理想低通作为逼近滤波器
wd1=boxcar(N)' ; b1=hd.*wd1;               % 用矩形窗设计
wd2=hanning(N)'; b2=hd.*wd2;               % 用汉宁窗设计
wd3=blackman(N)' ;b3-hd.*wd3;              % 用布莱克曼窗设计
[H1,w]=freqz(b1,1);                        % 用矩形窗设计的频率特性
[H2,w]=freqz(b2,1);                        % 用汉宁窗设计的频率特性
[H3,w]=freqz(b3,1);                        % 用布莱克曼窗设计的频率特性
subplot(1,2,1), plot(w,abs(H1),w,abs(H2),':',w,abs(H3),'-.');% 绘幅特性
legend('矩形窗','汉宁窗','布莱克曼窗')
subplot(1,2,2),            % 绘分贝幅特性
plot(w,20*log10(abs(H1)),w,20*log10(abs(H2)),':',w,20*log10(abs(H3)),'-.');
legend('矩形窗','汉宁窗','布莱克曼窗')
```

运行此程序所得三种滤波器的幅特性曲线见图 7.3.7。左图是幅特性的绝对值，右图则是用分贝表示，它的好处是能明显的分辨幅度很小的阻带衰减。

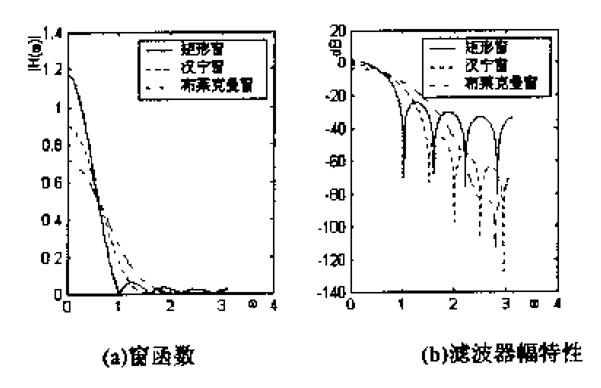


图 7.3.7　例 7.3.4 中用三种窗函数设计出的滤波器幅特性

从图 7.3.7 可以看出：用矩形窗时过渡带最窄，而阻带衰减最小；布莱克曼窗过渡带最宽，但换来的是阻带衰减加大。为了减少过渡带，必须加大窗口长度 N。下面给出利用窗函数技术和 MATLAB 子程序进行滤波器设计的几个实例。

例 7.3.5：根据下列技术指标，设计一个数字 FIR 低通滤波器：

$\omega_p=0.2\pi, R_p=0.25$ dB；$\omega_s=0.3\pi, A_s=50$ dB

从表 7.3.2 中选出一个适当的窗函数，确定脉冲响应，并给出所设计的滤波器的频率响应图。

解：哈明窗和布莱克曼窗均可提供大于 50 dB 的衰减。我们选择哈明窗，它提供了较小的过渡带，因此，具有较小的阶数。尽管在设计中，用不到通带波动值 R_p=0.25 dB，但必须检查设计的实际波动，验证它是否确实在给定容限内。下面的程序 hc735 给出了设计的步骤。

```
wp = 0.2*pi; ws = 0.3*pi; deltaw= ws - wp;  % 过渡带宽Δω的计算
N0 = ceil(6.6*pi/ deltaw);                   % 按哈明窗(表7.3.1)计算所需的滤波器长度N0
 N=N0+mod(N0+1,2);                           % 为了实现第一类偶对称滤波器，应确保其长度N为奇数
wdham = (hamming(N))';                       % 求窗函数
wc = (ws+wp)/2;                              % 截止频率取为两边缘频率的平均值
hd = ideallp(wc,N);                          % 求理想脉冲响应
h = hd .* wdham;                             % 设计的脉冲响应(系数)为理想脉冲响应与窗函数乘积
[db,mag,pha,grd,w] = myfreqz (h,[1]);        % 对设计结果进行检验
dw = 2*pi/1000;
Rp = -(min(db(1:wp/dw+1)))                   % 检验通带波动
As = -round(max(db(ws/dw+1:501)))            % 检验最小阻带衰减
% 以下为绘图程序，依次在四个子图中画出hd,wdham,h序列及分贝幅特性曲线。
```

在这个程序中，有几条语句需要解释一下：

（1）N=N0+mod(N0+1,2); 的作用是保证 N 为奇数，使滤波器始终保持类型 I 的性质。当 N0 为偶数时，mod(N0+1,2)=1，N=N0+1 就变为奇数；当 N0 为奇数时，mod(N0+1,2)=0，N=N0+0 仍为奇数。

（2）freqz 函数中把 0 到 2π 分为 1000 份，dw=2π/1000 是频率分辨率，Rp = -(min(db(1:wp/dw+1)))在下标 1 到通带下标 wp/dw+1 的范围内找分贝幅特性的最小值，即通带波动值；

（3）As = -round(max(db(ws/dw+1:501))) 是在阻带下标 ws/dw+1 到最高数字频率 pi 下标(501)的范围内找分贝幅特性的正整数最大值，即阻带最大衰减值。

计算结果如下：滤波器的长度 N=67，阻带衰减为 As=52 dB，达到了要求。对通带波动的验证显示，Rp=0.0394 dB。它是满足要求的。时域和频域的曲线如图 7.3.8 所示。

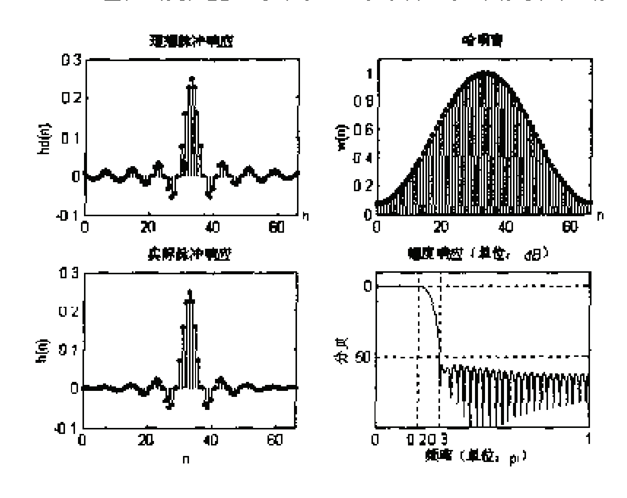


图 7.3.8　例 7.3.5 设计的 FIR 滤波器的系数（脉冲响应）及其频率响应

例 7.3.6： 利用例 7.3.5 给出的设计技术指标，选择凯泽窗，设计出所需的低通滤波器。

解： 设计步骤由下面的 MATLAB 程序 hc736 给出。

```
wp = 0.2*pi; ws = 0.3*pi; As = 50;
deltaf= (ws - wp)/(2*pi);                  % 过渡带宽 Δ f 的计算
N0 = ceil((As-7.95)/(14.36* deltaf))+1;    % 按(7.3.16)式计算所需的滤波器长度 N0
N=N0+mod(N0+1,2);                          % 为了实现第 I 类滤波器，应确保其长度 N 为奇数
beta = 0.1102*(As-8.7)                     % 按(7.3.17)式计算凯泽窗的 β 值
wdkai = (kaiser(N,beta))';                 % 求凯泽窗函数
wc = (ws+wp)/2;    hd = ideallp(wc,N);     % 求理想脉冲响应
h = hd .* wdkai;                           % 设计的脉冲响应(即系数)为理想脉冲响应与窗函数乘积
%检验语句同 hc735 最后带下画线的四条语句
% 以下为绘图程序，依次在四个子图中画出 hd, wdkai, h 序列及分贝幅特性曲线。
```

计算结果为：凯泽窗的参数为 $N=61$，$\beta=4.5513$，实际的阻带衰减为 52 dB。时域和频域的曲线在图 7.3.9 中给出。可见它比哈明窗具有较低的阶次，即可以用较少的运算时间，而得到同样的指标。

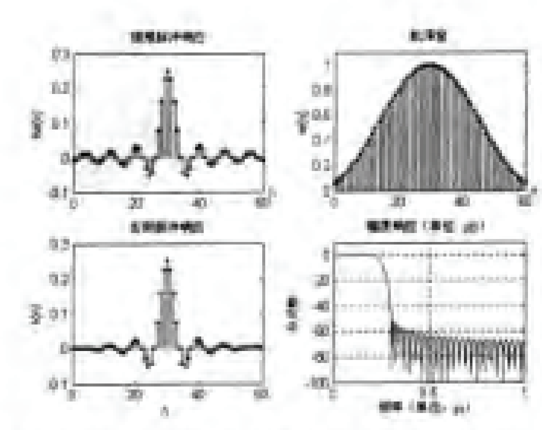


图 7.3.9　例 7.3.6 设计的 FIR 滤波器的系数（脉冲响应）及其频率响应

例 7.3.7： 设计下面的数字带通滤波器，其参数如图 7.3.10 所示。

低端阻带边缘：　$\omega_{1s}=0.2\pi$，　$A_s=60$ dB

低端通带边缘：　$\omega_{1p}=0.35\pi$，　$R_p=1$ dB

高端通带边缘：　$\omega_{2p}=0.65\pi$，　$R_p=1$ dB

高端阻带边缘：　$\omega_{2s}=0.8\pi$，　$A_s=60$ dB

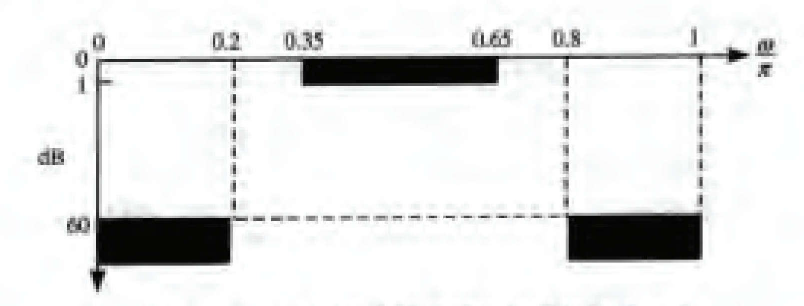


图 7.3.10　例 7.3.7 中的带通滤波器的技术指标

解： 存在两个过渡带，即 $\Delta\omega_1=\Delta\omega_{1p}-\Delta\omega_{1s}$，和 $\Delta\omega_2=\Delta\omega_{2s}-\Delta\omega_{2p}$，在窗设计中，这两个带宽必须相同。也就是说，对 $\Delta\omega_1$ 和 $\Delta\omega_2$ 不进行独立控制，因此 $\Delta\omega_1=\Delta\omega_2=\Delta\omega$。在此设计中，既可以用凯泽窗，也可以用布莱克曼窗。这里选用布莱克曼窗。同时也需要理想带通滤波器

响应 $h_d(n)$。此脉冲响应可以由两个理想低通幅度响应相减得到，如图 7.3.6 所示。下面的 MATLAB 程序 hc737 给出了设计步骤。

```
ws1 = 0.2*pi; wp1 = 0.35*pi;
wp2 = 0.65*pi; ws2 = 0.8*pi; As = 60;
deltaw= min((wp1-ws1),(ws2-wp2));            % 求两个过渡带中的小者
N0 = ceil(11*pi/deltaw);                     % 按表 7.3.2 中数据求滤波器应有长度 N0
N=N0+mod(N0+1,2);                            % 为了实现第一类偶对称滤波器，应使其长度 N 为奇数
wdbla = (blackman(N))';                      % 求窗函数
wc1 = (ws1+wp1)/2; wc2 = (wp2+ws2)/2;        % 截止频率取通带阻带边缘频率的平均值
hd = ideallp(wc2,N) - ideallp(wc1,N);        % 按图 7.3.6 求带通滤波器理想脉冲响应
h = hd .* wdbla;                             % 求实际滤波器脉冲响应，即其系数向量
%检验语句同 hc735 最后带下画线的四条语句
% 画图语句从略
```

计算结果为：布莱克曼窗的参数为 $N=61$，实际的阻带衰减为 75 dB。通带波动 R_p=0.0030。时域和频域的曲线在图 7.3.11 中给出。

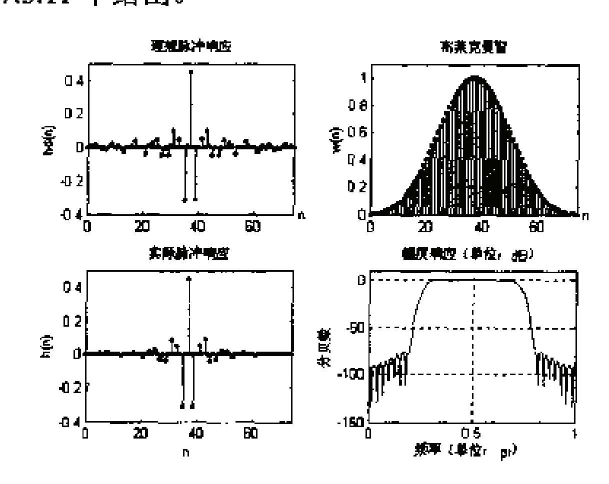


图 7.3.11　例 7.3.7 设计的 FIR 带通滤波器的系数（脉冲响应）及其频率响应

在信号处理工具箱中，MATLAB 提供了一个子程序叫做 fir1，它利用窗函数方法设计 FIR 滤波器，其标准调用格式为：

b=fir1(M,wn,'type',window)

其中：b 为待设计的滤波器系数向量，其长度为 $N=M+1$；

M 为所选的滤波器阶数；

wn 为滤波器给定的边缘频率，可以是标量，也可以是一个数组；

'type'为滤波器的类型，如高通、带通、带阻等，缺省时为低通；

window 为选定的窗函数类型，缺省时为 Hamming 窗。

例如 b=fir1 (32,0.7,'high',kaiser(33,4))表示用 β=4 的凯泽窗，设计阶数为 32（长度为 33），边界数字频率为 0.7π 的高通滤波器。键入这个语句，MATLAB 将给出设计的滤波器的系数向量 b。

在学习阶段不主张读者采用这种高度黑箱化的方法，因为这不利于掌握窗函数设计法的基本原理和构思，但在工程设计时调用这种函数则是无可非议的。高年级同学处在理论学习到工程实践的转换边界，对这类工具函数有所了解应该是有益的。

7.4 利用频率样本法设计 FIR 滤波器

窗函数法是从时域出发的一种设计方法。它从理想频率特性出发，用该特性的傅里叶反变换作为滤波器系数，由于要使它可以实现，并改善它的特性而加窗截断，使实际滤波器的频率响应偏离理想值，产生了通带波动、阻带衰减和过渡带。自然会产生一个问题，是不是可以不绕弯子，先在频域修改滤波器的预期频率特性，使它不要太理想，接近于实际的需要和可能，然后直接用它的傅里叶反变换作为滤波器系数？答案是肯定的。因为这种设计方法要用 IDFT，预期频率特性不是以连续函数的形式，而是在一组离散的频点上给出它的样本值，所以这种设计方法称为频率样本法。

7.4.1 预期频率特性的设置方法

由于希望把预期频率特性的傅里叶反变换作为滤波器系数，在设定预期频率特性时就必须根据下列几条基本原则：

（1）预期频率特性的样本点数应等于滤波器的长度 N，并在单位圆上等间隔分布；

（2）作为复数序列的预期频率特性应具有共轭对称性，以保证其傅里叶反变换所得系数为实序列；因此其幅特性应为偶函数，相特性应为奇函数；

（3）预期频率特性的相位特性应该与频率成线性关系；这意味着其符幅特性及其反变换所得序列应该具有对称或反对称的特点（见 5.5.2 节及 7.2 节）。

以上的原则可用数学公式描述如下。设滤波器预期频率特性用 $H_d(\omega)$ 表示，在 ω=0 到 2π 之间对它等间隔采样 N 点，得到：

$$H(k)=H_{\mathrm{d}}(\omega)\big|_{\omega=2k\pi/N}\quad k=0,1,\cdots,N-1 \tag{7.4.1}$$

用符幅特性及线性相位表示时，（7.4.1）式成为

$$H(k)=A(k)\mathrm{e}^{\mathrm{j}\theta(k)} \tag{7.4.2}$$

为了使(7.4.2)式的反变换 $h(n)$为实序列，且是满足 $h(n)=h(N-n-1)$的第 I 类或第 II 类线性相位滤波器，根据表 7.2.1 可以查到，$H(k)$的相特性 $\theta(k)$和符幅特性 $A(k)$应满足如下条件。

线性相位条件：

$$\theta(k)=-\frac{N-1}{2}\cdot\frac{2\pi}{N}k=-\frac{N-1}{N}k\pi \tag{7.4.3}$$

符幅特性的对称（或反对称）条件：

$$N\text{ 为奇数时}\qquad A(k)=A(N-k-1) \tag{7.4.4}$$

$$N\text{ 为偶数时}\qquad A(k)=-A(N-k-1) \tag{7.4.5}$$

注意不要把幅特性的对称性与系数 $h(n)$的对称性相混淆。

如果 $h(n)$是第 III 类或第 IV 类线性相位滤波器，也可从表 7.2.1 中找出对应的线性相位条件和符幅特性对称条件，此问题留给读者去思考。

在得知 $H(k)$, k=0,1, …, N−1 后，就可以用离散傅里叶反变换求出滤波器的脉冲响应 h(n)，也即是它的系数向量 $b(n)$。

$$b(n)=h(n)=\mathrm{IDFT}[H(k)] \tag{7.4.6}$$

7.4.2 频率样本法的设计过程

给定理想低通滤波器 $H_d(\omega)$，先选择滤波器长度 N，然后对 $H_d(\omega)$在 0 到 2π 上的 N 个等

间隔频率上采样，得到 $H(k)$，如图 7.4.1 左图所示。脉冲响应由离散傅里叶反变换式（7.4.6）得到。这种反变换在一定意义上也隐含了对样本 $H(k)$进行内插而得到实际响应 $H(\omega)$的过程。在第 6 章 6.3.5 节中曾指出，如果知道了单位圆上的 N 个样本点的频率特性，就可以求出系统在整个 z 平面传递函数。内插公式为第 6.3 节的(6.3.16)式：

$$H(z)=\frac{1-z^{-N}}{N}\sum_{k=0}^{N-1}\frac{H(k)}{1-W_N^{-k}z^{-1}}$$

这是频率样本法的理论基础。现在遇到的问题是由单位圆上的样本值求整个单位圆各处的频率特性，如从图 7.4.1 的左图求得右图，那就要先用 IDFT，由预期频率特性 $H(k)$求出系数 $h(n)$，再对 $h(n)$求 DTFT 得出实际的完整的频率特性 $H(\omega)$。

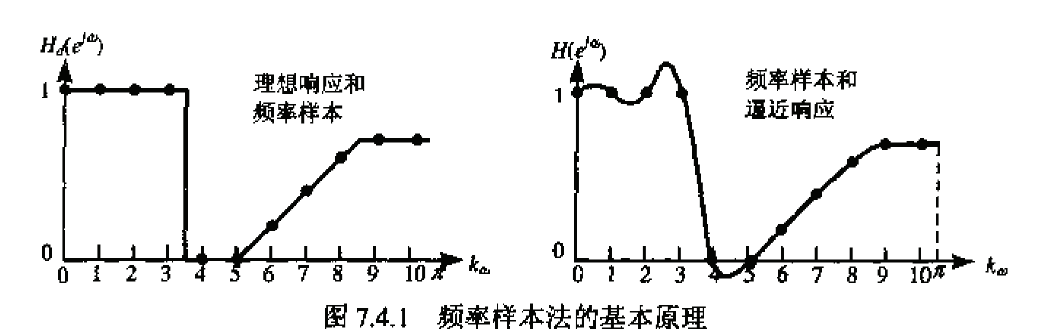


图 7.4.1　频率样本法的基本原理

由图 7.4.1 右图，可知频率样本法的特点是：

（1）在采样频率上的逼近误差为零，也就是理想的和实际响应的差为零；

（2）其余频率上的逼近误差取决于理想响应的形状：理想响应的轮廓越陡，则逼近误差越大；

（3）靠近带的边缘的误差大，在带内的误差小。

目前有两种设计方法，第一种直接用上面的基本思想，对逼近误差不加任何限制；也就是说，无论设计所得的误差为多大，我们都接受，这种方法叫做朴素设计法。第二种方法则通过改变过渡带的样本值，努力使阻带中的误差极小化，以便产生一个较好的设计，这叫做最优设计方法。

需要特别注意的是所取的样本点 $H(k)$必须是覆盖整个单位圆的 N 个样本点，不能少，也不能多。图 7.4.1 上只画了 0～π 的半个频段，这是因为对称性，在观察频率特性时我们已习惯了。但作反变换时，必须在 0～2π 的全频段上给出 $H(k)$。其次要注意，取了 ω=0 的点，就不能再取 ω=2π 点；同时要确保 $H(k)$对 N/2 处(也即对 ω=π 点)的循环对称性，否则就要出错。下面将用实例说明。

例 7.4.1：利用频率样本法设计线性相位低通滤波器，要求截止频率 ω_c=π/3，采样点数取 N=13 和 N=65 做比较。

解：用理想低通作为逼近滤波器。它在主值频域的幅特性及相特性如图 7.4.2 实线所示，对它的采样情况如图中圆点所示。采样得到的数据为(用 MATLAB 语言表达)：

```
A=[ones(1,3),zeros(1,8),ones(1,2)];     (见图7.4.2)
theta = - pi*[0:12]*12/13;              (按公式 (7.4.3))
```

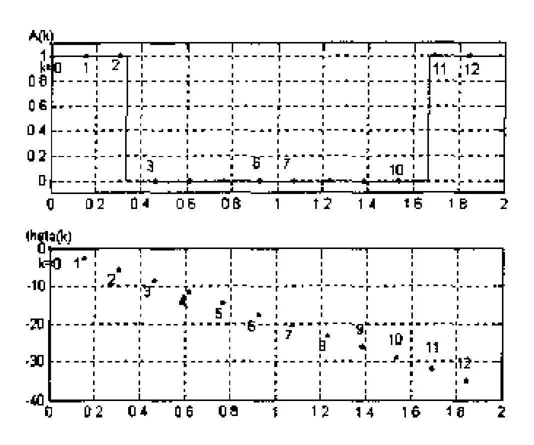


图 7.4.2　理想低通滤波器频率特性的 13 点样本序列

要特别注意这 13 个点中包括了 $\omega=0$，而不包括 $\omega=2\pi$。它把整个单位圆等分为 13 份，样本点就不多不少地均匀分布在这 13 个位置上。符幅特性对于 k=13/2=6.5(也即 $\omega=\pi$)处循环对称，从循环的意义上，k=0 的点可以看做分属最左和最右，所以 A 序列左边三个 1，右边两个 1 并不影响它的循环对称性。theta 对 k = 6.5 处的奇对称性是由式（7.4.3）保证的，通常不必检验，也不必画图。对由 A 和 theta 构成的符幅特性进行 IDFT，就可得到 h(n)。然后计算其频率响应。所用的 MATLAB 程序 hc741 如下。编程中考虑了对不同 N 值有通用性。

```
N=input('N= '); wc=pi/3;
N1= fix(wc/(2*pi/N)); N2=N-2*N1-1;  % 样本间距为 2*pi/N, N1 为通带 wc 的样本数
A=[ones(1,N1+1),zeros(1,N2),ones(1,N1)];        % 符幅特性样本序列, N2 为阻带长度
theta = - pi*[0:N-1]*(N-1)/N;                   % 相位特性样本序列
H=A.*exp(j*theta);                              % 频率特性样本序列
h=real(ifft(H));                                % 反变换求出脉冲序列, 去掉运算误差造成的虚部
[db,mag,pha,grd,w] = myfreqz (h,[1]);           % 检验设计好的滤波器的频率响应
%检验语句同 hc735 最后带下画线的四条语句
% 以下为画图程序, 画出设计完的滤波器的幅特性
```

在 N=13 的条件下运行此程序，将结果画入四个子图。第一个子图中画理想幅频特性及取的样本；第二个子图中画根据这些样本反变换而得的滤波器系数向量；第三个子图中画实际滤波器的幅频特性；第四个子图中画实际滤波器的分贝幅频特性如图 7.4.3 所示。数值结果为：Rp=1.0535 dB，As=16 dB。

在 N=13 和 N=65 两种条件下运行此程序，并将其幅特性画在同一张图上，得到图 7.4.4。该图表明：

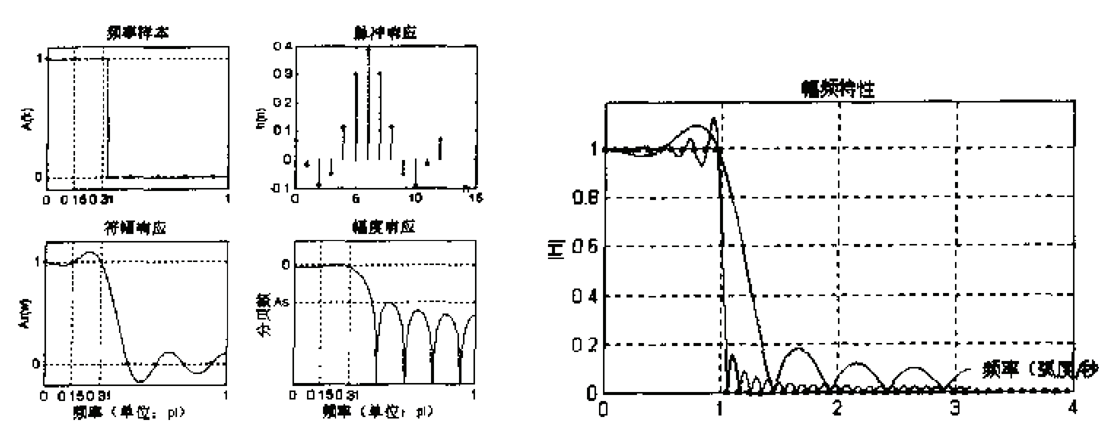


图 7.4.3　频率样本法 N=13 时滤波器幅特性　　图 7.4.4　频率样本法 N 取不同值时滤波器幅频特性

（1）两种情况下样本点处的幅特性都准确地等于给定值 1 或 0，在样本点之间的区域会有一定误差。所以，N 愈大，给出的样本点愈多，设计出的滤波器愈接近理想特性；

（2）在 k=N1/N 和(N1+1)/N 之间出现过渡带。当 N=13 时，过渡带宽为 $\pi/13$，Rp= 1.0535。注意到其最小阻带衰减 As 为 16 dB，显然是不能接受的。而 N=65 时，过渡带宽为 $2\pi/65$。过渡带宽愈窄，因吉布斯效应产生的通带波动愈剧烈，Rp= 1.6733，As=18 dB 这两个指标基本没有改善；

（3）要想又把 N 取大，又不让过渡带宽过分减小，就要避免频率特性中的突跳，这只有在过渡带中放一个或几个样本点才行。

优化设计法：为了得到更大的衰减，必须增大 N，并且使过渡带中的样本成为自由样本，也就是说，可以改变它们的样本值，以便在给定的 N 和过渡带宽下得到最大的衰减。这个问题称为优化问题，利用线性规划技术可以解决它。若不想涉及优化的数学问题，也可以用试凑方法来解决，下面的例子可以说明改变过渡带样本值对设计的影响。

例 7.4.2：用加宽过渡带的方法，为例 7.4.1 设计一个较好的低通滤波器。

解：先考虑在过渡带中增加一个样本，把阻带边界上的脉冲由零改为 T1。这样，程序 hc741 中的符幅特性赋值语句就成为：

```
A=[ones(1,N1+1),T1,zeros(1,N2-2),T1,ones(1,N1)];          % 符幅特性样本序列
```

按 N=65 的情况，把 k=11 和 k=55 处的符幅特性样本值设为 T1。另外要改的是阻带的第一个样本点 ws1，要把它比原来向右移一个采样间距 2*pi/N:

```
ws1=wp1+2*2*pi/N;
```

还要加一条对 T1 的赋值语句。为了进行试凑，可把它写成键盘输入形式：

T1=input('过渡带中单个样本值 T1');

其他语句不变。这样构成的程序为 hc742a，已放在程序盘中，此处不再列出。

运行此程序并依次输入 T1=0.5,0.4,0.3 可得到表 7.4.1 中的结果。

表 7.4.1　过渡带中加一个样本时样本值对通带波动和阻带衰减的影响

T1	0.5	0.4	0.3	0.38
Rp	0.3854	0.6467	0.9059	0.6987
As	30	40	36	44

继续输入两位有效数的 T1 值进行试凑，可知 T1 取 0.38 时，可以得到最大的阻带衰减 44 dB。

进一步考虑在过渡带内放两个幅度样本，这样过渡带的宽度达到了三个采样间距。假如这两个过渡带样本，一个占用原来通带的末样本的位置，另一个占用阻带的首样本位置，则符幅序列的赋值语句成为：

```
A=[ones(1,N1),T1,T2,zeros(1,N2-2),T2,T1,ones(1,N1-1)];          % 符幅特性样本序列
```

同时，还要改通带和阻带的频率范围语句

```
wp1=2*pi/N*fix(pi/3/(2*pi/N) -1);          % 通带的末样本点 wp1
ws1=wp1+3*2*pi/N;                          % 阻带的首样本点 ws1
```

还要加两条分别对 $T1$，$T2$ 的赋值语句。为了进行试凑，可把它写成键盘输入形式：

```
T1=input('过渡带中第一个样本值 T1');
T2=input('过渡带中第二个样本值 T2');
```

其他语句不变。这样构成的程序为 hc742b，已放在程序盘中，此处不予列出。

运行此程序并依次输入 $T1$，$T2$ 的一系列两位有效数进行试凑，可得到表 7.4.2 的结果。

表 7.4.2　过渡带中加一个样本时样本值对通带波动和阻带衰减的影响

$T1$	0.59	0.60	0.58	0.59	0.59
$T2$	0.11	0.11	0.11	0.10	0.12
Rp[dB]	0.3176	0.2918	0.3433	0.3025	0.3326
As[dB]	65	59	59	57	57

从此表可以看到 $T1$=0.59，$T2$=0.11 是一个局部最优的组合，此时可以得到最大的阻带衰减 65 dB。这个条件下画出的与图 7.4.3 相应的曲线见图 7.4.5。

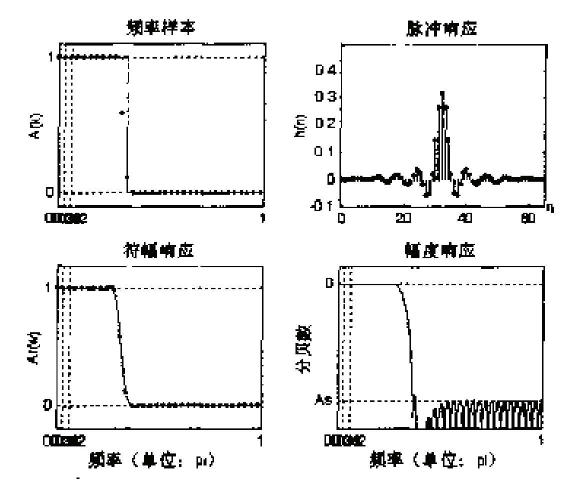


图 7.4.5　过渡带取两个最优样本，N=65 时滤波器幅特性

频率样本法设计滤波器最大的优点是直接从频率域进行设计，比较直观，也适合于设计具有任意幅度特性的滤波器。缺点是边缘频率不易控制。如果增加采样点数 N，对确定边缘频率有好处，但 N 加大会增加滤波器的成本。因此，它适合于窄带滤波器的设计。

7.4.3　MATLAB 中的频率样本法设计函数

MATLAB 信号处理工具箱提供了一个频率样本法的设计函数 fir2。它的典型调用方法为：h =fir2(M, f, A)。

其中 M 为 FIR 数字滤波器的阶数（滤波器的长度为 $N=M+1$），它的预期频率响应在数组 f 和 A 中给定。长度为 N 的数组 h 是返回的滤波器系数（或脉冲响应）。数组 f 中包含各边缘频率，其单位为 π，即 $0.0 \leqslant f \leqslant 1.0$。$f$=1 对应于采样频率的一半，即奈奎斯特频率。这些频率必须以递增次序排列，从 0 开始，到 1 结束。数组 A 为各指定频率上预期的幅度响应。f 和 A 的长度必须相等，plot(f,A)应该给出预期的滤波器幅频特性。

fir2 函数缺省地使用 Hamming 窗。所以实际得到的系数序列是频谱的傅里叶反变换再乘以窗函数。可以加第四个变元来换成其他的窗函数，如 h =fir2(M,f,A, boxcar(M+1))就指定使

用长度为 $N=M+1$ 的矩形窗。

fir2 函数的一个很有用的功能是可以按指定频谱形状，生成信号序列，如例 7.4.3 所示。

例 7.4.3： 要求生成一个限带序列 $h(n)$，其长度为 51，它的频谱应该如图 7.4.6 上图。

解： 先按预期的频率特性写出 f, A 向量，然后调用 fir2 函数，用下列程序 hc743 来完成。

```
N=input('N=');M=N-1;                       % 给定序列长度
f=[0,0.45,0.5,1]; A=[0,1,0,0];             % 给定预期频谱特性
h=fir2(M,f,A);                             % 生成脉冲响应序列
[H,w]=freqz(h,1);                          % 计算脉冲序列的频谱
```

绘图语句略去

运行该程序，键入 N=51,得到的结果见图 7.4.6。可以看出，生成的序列具有很接近预期形状的频谱。所用的语句又非常简明，是一个不错的方法。

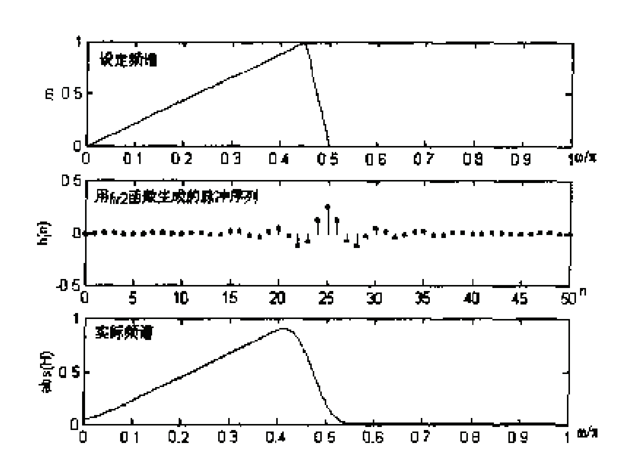


图 7.4.6　按照给定频谱（上），用 fir2 函数生成序列（中）及该序列的实际频谱（下）

7.5　FIR 滤波器的最优设计法

上面探讨的两种技术，即窗函数设计和频率样本技术，都易于理解和实现。但是，它们存在一些不足：第一，在设计过程中，不能精确指定通带和阻带频率 ω_s 和 ω_p，只能接受设计所得的大体合用值；第二，不管是在窗设计方法中使 $\delta_1=\delta_2$，还是在频率样本法中只优化 δ_2，均不能同时控制波动系数 δ_1 和 δ_2；第三，理想响应和实际响应之间的逼近误差，在全频带区间上不是均匀分布的。靠近边缘频率处误差较高，而在远离边缘频率处较小。可以设想，如果使误差均匀分布，就可以在满足相同技术指标的条件下，得到一个较低阶滤波器。最优化就是一种能克服上述三个问题的数学方法。相对来讲，这种方法用的数学较深，不过现在可以依靠计算机中编好的子程序，所以已不算什么难题了。

最优化的思想实际上在窗函数法中就有反映。它用窗函数直接截取理想滤波器脉冲响应 $h_d(n)$ 的一段，作为滤波器的 $h(n)$，这是一种时域逼近法。用 $E(\omega)$表示理想滤波器 $H_d(\omega)$和所设计滤波器 $H(\omega)$之间的幅频特性误差

$$E(\omega)=H_d(\omega)-H(\omega) \tag{7.5.1}$$

如果以这个误差平方的连续积分

$$e^2=\frac{1}{2\pi}\int_{-\pi}^{\pi}[E(\omega)]^2 d\omega \qquad (7.5.2)$$

作为最优化的目标函数，则可以证明采用矩形窗时，均方误差是最小的，也就是说，矩形窗是连续最小均方误差的最优设计法。注意：这里最小是指在整个频带上积分最小，它保证了误差平方所包的面积最小，但不能保证误差幅度最小。由于吉布斯效应，使过渡带附近的通带内有较大的上冲，阻带旁瓣也不小。换句话说，频域的连续最小均方误差设计法的结果并不理想，所以实际上很少使用矩形窗。可见这种连续最小均方误差设计法没有得到应用。

本节将介绍两种最优化方法，第一种是离散最小二乘法。它的思路是使在给定的一些离散频点上，使实际幅频特性与理想幅频特性之间的误差的平方和为最小；第二种是最小最大波动法，也称为等波动法。具有这种性质的滤波器幅特性在通带和阻带上的误差峰值是均匀分布的。也就是说，其误差具有等波动特性。因而可把波动的幅度控制到最小，或在同等指标下减小它的阶次。第一种方法是连续最小平方积分法的推广，容易理解，但它的指标没有和滤波器的指标挂钩，误差平方积分小的滤波器不能保证没有窄而大的波动出现，像吉布斯效应那样。而第二种方法则直接控制了通带波动和阻带衰减，所以最具有针对性，滤波器的最优设计主要指的是这种方法。但是它用到的数学超过了大学本科的水平，所以在本书中只能概要的介绍一下它的思想。好在有 MATLAB 信号处理工具箱提供的函数帮助，不掌握它的数学推导并不影响使用这种方法。

7.5.1 设计 FIR 滤波器的离散最小二乘法

这里讨论的最小二乘法是对滤波器幅特性而言的。从频率样本法出发，先分析实际幅特性与给定的（理想的）幅特性之间的误差，求出它们的平方和与（7.5.2）式中 e^2 的不同在于，这里用的是有限个指定样本点上的误差平方和，而不是连续积分。然后和滤波器的设计参数建立数学关系，最后找到能使指标达到最小的参数的公式。

以类型 I 的线性相位滤波器为例，因为 N 为奇数，$L=\tau$，它的频率特性具有如下所示的第 7.2 节式（7.2.2 a）的形式。

$$H(e^{j\omega})=\left[\sum_{n=0}^{L}2h(n)\cdot\cos(\omega(L-n))-h(L)\right]e^{-j\omega L}=A(\omega)e^{-j\omega L}$$

其相位特性$-\omega L$ 与 ω 成线性关系，故幅特性部分具有正负号，所以符幅特性具有如下所示的第 7.2 节式（7.2.2 b）的形式。

$$A(\omega)=\sum_{n=0}^{L}2h(n)\cdot\cos[\omega(L-n)]-h(L) \qquad \omega=[0\sim2\pi]$$

如果滤波器系数向量，也就是它的脉冲响应长度为 N（对 I 类滤波器，N 为奇数），由于存在着对称性的限制，可以独立选择的系数不是 N 个，而是 $L+1=(N+1)/2$ 个。从(7.2.2 b)也可以看出这点，为了更明确和更普遍起见，可以把它写成：

$$A(\omega)=\sum_{n=0}^{L}d(n)\cdot\cos n\omega \qquad \omega=[0\sim2\pi] \qquad (7.5.1)$$

其中：

$$d(n)=\begin{cases}2h(L-n) & n=1,2,\cdots,L\\ h(L) & n=0\end{cases} \qquad (7.5.2)$$

因此滤波器的符幅特性就像傅里叶级数那样，由 L 个余弦波和一个直流分量组成。要调

整它的系数向量 $d(n)$，就是要调整其脉冲响应的各个分量 $h(n)$，最后可以近似地拟合出我们需要的理想（或预期）幅特性。对于类型 II, III 和 IV 线性相位滤波器，其符幅特性都能化为(7.5.3)式的形式，只是 $d(n)$的形式不同，或把余弦换成正弦。因此可以把这四种类型的滤波器归纳为统一的形式(7.5.3)式来表示，只是其中的 $d(n)$与滤波器系数 $h(n)$的关系不同，推导出的结果参阅表 7.5.1。其共同特征是靠多个谐波来拟合幅特性。因此以(7.5.3)式进行讨论就具有代表意义。

表 7.5.1 四类线性相位 FIR 滤波器的符幅特性

类型	系数长度 N 奇偶 对称性（奇偶）	$A(\omega)$	系数关系
I	N 奇数, 偶对称 $h(n)=h(N-n)$	$\sum_{n=0}^{L} d(n)\cos n\omega$	$d(0)=h(L)$ $d(n)=2h(L-n)$ $n=1,2,\ldots,L$ $h(n)=d(L-n)/2$ $n=0,1,\ldots,L-1$ $h(L)=d(0)$
II	N 偶数, 偶对称 $h(n)=h(N-n)$	$\sum_{n=0}^{L} d(n)\cos[(n+0.5)\omega]$	$d(n)=2h(L-n)$ $n=0,1,\ldots,L$ $h(n)=d(L-n)/2$ $n=0,1,\ldots,L$
III	N 奇数, 奇对称 $h(n)=-h(N-n)$	$\sum_{n=0}^{L} d(n)\sin\omega$	$d(0)=0$, $h(L)=0$ $d(n)=2h(L-n)$ $n=1,2,\ldots,L$ $h(n)=d(L-n)/2$ $n=0,1,\ldots,L-1$
IV	N 偶数, 奇对称 $h(n)=-h(N-n)$	$\sum_{n=0}^{L} d(n)\sin[(n+0.5)\omega]$	$d(n)=2h(L-n)$ $n=0,1,\ldots,L$ $h(n)=d(L-n)/2$ $n=0,1,\ldots,L$

注：滤波器系数向量长度为 N，滤波器阶次为 N-1，$L=\lfloor (N-1)/2 \rfloor$，独立参数数目为 L+1

为了推导公式的方便，以后将预期频率特性的符幅特性用 $D(\omega)=|H_d(\omega)|$表示。即

$$H_d(\omega) = D(\omega)e^{-j\omega\tau}$$

假如给定一个滤波器的预期符幅频性 $D(\omega)$在 $\omega=\omega_i (i=1, 2, \cdots, K)$的 K 个频点上的值，那么把实际滤波器的符幅特性与预期特性相拟合的方程有 K 个：

$$A(\omega_i) = \sum_{n=0}^{L} d(n)\cos n\omega_i \approx D(\omega_i) = D_i \qquad (i=1, 2, \ldots, K) \tag{7.5.3}$$

由于 $A(\omega_i)$ 是 $L+1$ 个谐波之和，这方程中的待定参数有 $L+1$ 个。

如果 $K=L+1$，即方程数与未知数的数目相等，那么由这些方程恰好解出了这些系数。上节的频率样本法大体就近似于这种情况。那时是用 IFFT 由预期频率特性计算脉冲响应的，由 IFFT 的特性，频域的样本点数与时域序列的长度相同。虽然由于对称性，实际的独立参数数目只有一半，但同样由于频域的对称性，给定的方程也只有一半有效。不同点是由于傅里叶反变换的要求，样本的频点必须等间隔分布。而用最小二乘法时，频点的数目和位置都完全可以自由选择。

如果 $K>L+1$，即方程数比未知数的数目多，形成了所谓超定方程组。那就不可能找到精确满足这些方程组的系数 $d(n)$，只能找到最近似地满足这些方程的最小二乘解。这个最小二乘解的含义是：把这些系数代入方程后，得到的幅特性与预期特性在样本点处的误差平方和为最小。由于未知数 $d(n)$在方程组中是以线性项的形式出现的，它的最小二乘解可以用解析

方法求得。实际工程中总是希望减少滤波器的阶次，也就是减少系数的数目，因为这可以降低滤波器的成本和运算时间。人们总想用最少的系数，在尽量多的频点上近似逼近预期幅特性。这所以种情况是最常见的。

如果 $K<L+1$，则形成了不定方程组，那会有无穷个解，工程上意味着浪费，是没有意义的。所以只考虑 $K>L+1$ 的情况，推导最小二乘解的公式。

因为方程组(7.5.3)无法精确满足，把它改写为如下的误差方程组：

$$e_i = D_i - \sum_{n=0}^{L} d(n)\cos n\omega_i \qquad (i=1,2,\cdots,K) \tag{7.5.4}$$

可以用矩阵形式表示为

$$e=D-Pd \tag{7.5.5}$$

其中：e 为单列 K 元误差向量，D 为预期幅特性在样本点列上的 K 元单列向量，d 为 $L+1$ 元待定系数单列向量，而 P 则是由 $\cos(n\omega_i)$组成的 $K\times(L+1)$的系数矩阵。

$$e=\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_K \end{bmatrix},\ D=\begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_K \end{bmatrix},\ P=\begin{bmatrix} 1 & \cos(\omega_1) & \cdots & \cos(L\omega_1) \\ 1 & \cos(\omega_2) & \cdots & \cos(L\omega_2) \\ & \vdots & & \\ 1 & \cos(\omega_K) & \cdots & \cos(L\omega_K) \end{bmatrix},\ d=\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_L \end{bmatrix}$$

最优化的目标函数（上角标 T 表示转置）是

$$J=\sum_{i=1}^{K} e_i^2 = e^T\cdot e=(D-Pd)^T(D-Pd)=D^TD-D^TPd-d^TP^TD+d^TP^TPd \tag{7.5.6}$$

使目标函数取极值的参数 d 应该满足 $\partial J/\partial d=0$，由此得到 $P^TPd-P^TD=0$ 或$P^TPd=P^TD$。最后得到

$$d=\left(P^TP\right)^{-1}P^TD \tag{7.5.7}$$

这就是最小二乘法求最优化滤波器系数的公式。其中 P^TP 为$(L+1)\times(L+1)$方阵，因而$\left(P^TP\right)^{-1}$也是$(L+1)\times(L+1)$的方阵，P^TD 为$(L+1)\times 1$ 单列矩阵，求出的 d 也为$(L+1)\times 1$ 单列矩阵。用 MATLAB 计算(7.5.9)式特别方便，可以表示为 d=inv(P'*P)*P'*D 也可以更简便地用 MATLAB 中定义的左除运算 d=P\D 来完成。

例 7.5.1：设一个长度为 N=9 的 I 型线性相位 FIR 滤波器，要使它在 0～π 之间的八个频点 ω 上逼近预期的低通幅特性 D：

ω=[0, 0.33, 0.67, 1, 1.5, 2, 2.5, 3.14]; D=[1,1,1,1,0,0,0,0];

解：先列出方程组(7.5.3)的完整形式：

$$\begin{bmatrix} 1 & \cos\omega_1 & \cos 2\omega_1 & \cos 3\omega_1 & \cos 4\omega_1 \\ 1 & \cos\omega_2 & \cos 2\omega_2 & \cos 3\omega_2 & \cos 4\omega_2 \\ 1 & \cos\omega_3 & \cos 2\omega_3 & \cos 3\omega_3 & \cos 4\omega_3 \\ 1 & \cos\omega_4 & \cos 2\omega_4 & \cos 3\omega_4 & \cos 4\omega_4 \\ 1 & \cos\omega_3 & \cos 2\omega_5 & \cos 3\omega_5 & \cos 4\omega_5 \\ 1 & \cos\omega_6 & \cos 2\omega_6 & \cos 3\omega_6 & \cos 4\omega_6 \\ 1 & \cos\omega_7 & \cos 2\omega_7 & \cos 3\omega_7 & \cos 4\omega_7 \\ 1 & \cos\omega_8 & \cos 2\omega_8 & \cos 3\omega_8 & \cos 4\omega_8 \end{bmatrix}\begin{bmatrix} d(0) \\ d(1) \\ d(2) \\ d(3) \\ d(4) \end{bmatrix}=\begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \\ D_7 \\ D_8 \end{bmatrix} \tag{7.5.8}$$

看来系数矩阵 P 的列写比较麻烦，其实，回想求离散傅里叶变换时的做法，利用列矩阵 w=[w_1, w_2, …; w_8]乘行矩阵[0:4]形成一个 8×4 矩阵，然后求余弦得到 P。这可以用 MATLAB 语句 P=cos(w'*[0:4])轻而易举地列出。因此用下列几行 MATLAB 程序 hc751 就可以方便地完成最小二乘最优滤波器的设计。

```
N=9; L=floor((N-1)/2);                     % 列出待求系数序数
w=[0,0.33,0.67,1,1.5,2,2.5,3.14];          % 列出频率向量
D=[1,1,1,1,0,0,0,0];                       % 列出预期幅特性向量
P=cos(w'*[0:L]);                           % 列出 P 矩阵
d=(P'*P)\P'*D'                             % 求出待求系数
```

程序运行的结果为

```
d =  0.3981
     0.6039
     0.2137
    -0.1205
    -0.1506
```

知道 $d(n)$以后，就可以按表 7.5.1 中第一行末栏的公式算出相应的滤波器系数 $h(n)$。变成行向量，写成下列语句

```
h(1:4)=d(5:-1:2)'/2;h(5)=d(1); h(6:9)=d(2:5)';
```

得到

```
h=[ -0.0753, -0.0603, 0.1069, 0.3020, 0.3981, 0.3020, 0.1069, -0.0603, -0.0753]
```

根据这个系数向量即可以计算滤波器的频率特性进行校核。熟练者也可以直接列出语句

```
h=[flipud(d(2:5)/2)',d(1),d(2:5)'/2]  % 由d求h
[H,w]=freqz(h,1);                      % 由脉冲响应求频率特性
plot(w,abs(H))                         % 画出实际的幅频特性
```

程序运行的结果见图 7.5.1。因为 L=4，最高的谐波次数为 4，在 0～π 之间幅特性摆动最多两个周期，达到峰值的次数应为四次，这从图上也看得很清楚。

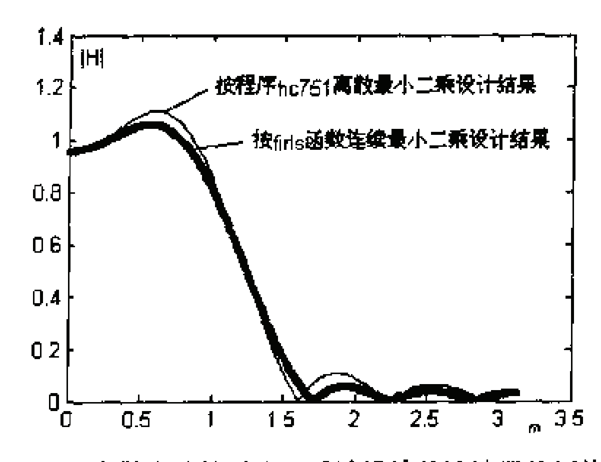


图 7.5.1　离散和连续最小二乘法设计的滤波器的幅特性

信号处理工具箱中提供了一个 FIR 滤波器最小二乘设计函数 firls，它的调用格式为

b=firls(M,f,A)

其中，M 为滤波器阶次，用本书的符号，$M=N-1$。f 为频率向量，A 为幅度向量，f 和 A 共同描述了预期频率特性。它们具有同样长度，都有偶数个元素。$f(1)$，$f(2)$之间的预期幅特性用 $A(1)$ 和 $A(2)$的连线表示；$f(3)$，$f(4)$之间的预期幅特性用 $A(3)$ 和 $A(4)$的连线表示，而$f(2)$、$f(3)$之间，$f(4)$、$f(5)$之间则看做过渡带，属于不关心频段，其频率特性不做限制。firls 函数对所有关心频段上的误差平方进行积分，并按此值为最小设计 FIR 滤波器的系数。因此它和本节介绍的方法思路相同，但在处理细节上有些区别。我们用的是离散的频率样本点上的误差

平方和，firls 则是在关心频段内等间隔地取密集样本点的误差平方和（相当于积分），所以它是一种指定频段上的连续的最小二乘优化法。可以设想，指定的频段是全频段时，它是前面说的效果不好的连续误差平方法；若指定的频段较多又短时，它就是离散最小二乘法。所以 firls 是连续最小二乘法与离散最小二乘法的概括。

用分段连续的最小二乘优化法时，例 7.5.1 的设计程序如下：

```
F=[0,1,1.5,pi]/pi,A=[1,1,0,0],          % 给出预期频率特性
b=firls(N-1,F,A),                        % 调用设计函数,求出滤波器系数
[H1,w1]=freqz(b,1);                      % 检验幅特性
plot(w1,abs(H)),hold off                 % 绘制幅频特性曲线
```

运行结果的系数向量为

```
b= -0.0568  -0.0506  0.0897  0.2964  0.3966  0.2964  0.0897  -0.0506  -0.0568
```

其幅特性也绘制在图 7.5.1 上。可见两种设计方法所得的结果是相近的。

7.5.2　最小最大波动法设计滤波器

从上节最小二乘法设计的幅特性曲线图 7.5.1 中，可以看到还有两方面的潜力：一是它在阻带内的最大波动在阻带边缘处最大，随频率的增加逐次减小，如果把阻带边缘处的波动削减一些补到高频区去，使这些波动变成一样大，那就可能使阻带衰减加大；第二，在计算误差平方时，它把通带内的波动和阻带内的波动同等看待进行最小化。实际上，大部分滤波器对通带内的波动容差比较大，幅特性为 1 和 1.01 的误差一般对系统的影响不大，而对于阻带幅特性 0.011 与 0.001 之差，虽然同是 0.01，泄漏进来的干扰信号却大了十倍，所以对它的要求就相当严格。如果我们放宽一些对通带中波动的要求，就可能在同样的滤波器阶数条件下，进一步降低阻带的最大波动指标。最小最大波动法的思想就是使全频段上的最大波动达到最小，而且通过加权使通带波动和阻带波动控制到规定的比例。人们为了这个目的进行了长期的研究，切比雪夫, Parks-McClellan, Remez 等学者分别从各自的学科对解决这个问题作了贡献，其综合成果就体现在本节中。所以这个方法有的地方称为切比雪夫逼近法，也有的称为雷米兹法，不太统一。

（1）切比雪夫最佳一致逼近准则

设待设计的滤波器预期幅度特性为 $D(\omega)$，实际的线性相位滤波器符幅特性 $A(\omega)$，其加权误差 $e(\omega)$用下式表示：

$$e(\omega)=Wt(\omega)[A(\omega)-D(\omega)]=Wt(\omega)\sum_{n=0}^{L}d(n)[\cos n\omega-D(\omega)] \tag{7.5.9}$$

式中，$D(\omega)$,$A(\omega)$,$e(\omega)$均同(7.5.3)式和(7.5.4)式，Wt（ω）称为误差加权函数，它是为在通带或阻带取不同的波动值而设置的。一般地，在要求逼近精度高的频带，$Wt(\omega)$取值大，要求逼近精度低的频带，$Wt(\omega)$取值小。设计过程中 Wt（ω）是由设计者取定的函数。对于低通滤波器，常取

$$Wt(\omega)=\begin{cases}\delta_2/\delta_1 & 0\leq\omega\leq\omega_p\\ 1 & \omega_p<\omega\leq\pi\end{cases} \tag{7.5.10}$$

其中 δ_1 和 δ_2 分别为滤波器指标中的通带和阻带的容许波动。如果 $\delta_2/\delta_1<1$，说明对通带波动的加权比较小。例如 $\delta_2/\delta_1=0.1$，用这样的加权设计的等波动滤波器，在通带内的最大波动 δ_1 将比阻带内最大波动 δ_2 大 10 倍。

设计问题的提法是：找到滤波器系数向量 $d(n)$，使得在 $\omega=[0,\omega_p]$ 和 $\omega=[0,\omega_p]$频段内的最大绝对误差$|e(\omega)|$为最小。这就称为最大误差最小化(Minimax)问题。这个问题在数学文献上很早就有记载，称为切比雪夫逼近问题。它的解也已经知道，那就是 $e(\omega)$ 在 $\omega=[0,\omega_p]$ 和 $\omega=[0,\omega_p]$频段内必须有正负交替出现且极值相等的波动，如图 7.5.3 所示。它可以用一个定理来正式地予以描述，称为交替定理。

交替定理：设 Ω 是在$[0,\pi]$上的一个闭子集，比如是$[0,\omega_p]$和$[\omega_s,\pi]$的合集。则

$$A(\omega)=\sum_{n=0}^{L}d(n)\cos n\omega \tag{7.5.11}$$

是对 $D(\omega)$的在最小最大意义下的最佳逼近，当且仅当(7.5.11)式的误差函数 $e(\omega)$是等波动的，并且该误差函数 $e(\omega)$在 Ω 中至少有 $L+2$ 个极值点。也就是说，在 Ω 中，存在着

$$0\leqslant\omega_1<\omega_2<\ldots\omega_{L+2}\leqslant\pi$$

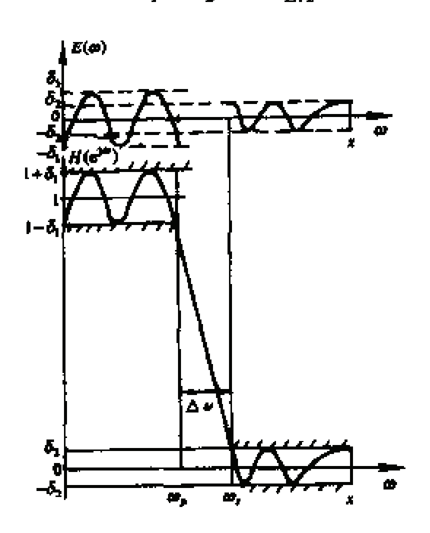


图 7.5.3　低通滤波器的最佳逼近

其中包括 ω_p和 ω_s，在这些频率上的误差函数正负交替，都达到同样的极值 e_m

$$e(\omega_i)=(-1)^i e_m \quad i=1,2,\cdots,L+2 \tag{7.5.12}$$

其中 e_m 可以是一个正数或负数，满足

$$|e_m|=\max_{[\omega\in\Omega]}|e(\omega)| \tag{7.5.13}$$

运用这个定理，就意味着在通带和阻带内的符幅特性分别满足：

$$|A(\omega)-1|\leqslant\left|\frac{\delta_1}{\delta_2}e_m\right|=\delta_1 \qquad 0\leqslant\omega_1\leqslant\omega_p \tag{7.5.14}$$

$$|A(\omega)|\leqslant e_m=\delta_2,\quad \omega_s\leqslant\omega\leqslant\pi \tag{7.5.15}$$

所以通带内的最佳 $A(\omega)$应在 $1\pm\delta_1$之间正负交替地波动，而阻带内的最佳 $A(\omega)$应在$\pm\delta_2$之间正负交替地波动，如图 7.5.4 所示。通过调整加权函数中的 $\delta1/\delta2$ ，就可以控制通带和阻带波动之间的比例。

现在来研究如何应用交替定理进行设计。因为它指出了至少有 $L+2$ 个或更多极值点，如果知道了 $L+2$ 个极值点的频率，那就足以解决问题了。把公式(7.5.11)代入(7.5.9)，然后再代入式(7.5.12)中，得到：

$$\sum_{n=0}^{L}d(n)\cos n\omega_i-(-1)^i\frac{e_m}{Wt(\omega_i)}=D(\omega_i)=D_i \qquad i=1,2,\cdots,L+2 \tag{7.5.16}$$

这里有 $L+2$ 个方程，未知数为 $L+1$ 个 $d(n)$再加上 e_m，总数也是 $L+2$ 个，因此方程有解。设 $L=4$，列出它的矩阵形式：

$$\begin{bmatrix} 1 & \cos\omega_1 & \cos 2\omega_1 & \cos 3\omega_1 & \cos 4\omega_1 & 1/Wt(\omega_1) \\ 1 & \cos\omega_2 & \cos 2\omega_2 & \cos 3\omega_2 & \cos 4\omega_2 & -1/Wt(\omega_2) \\ 1 & \cos\omega_3 & \cos 2\omega_3 & \cos 3\omega_3 & \cos 4\omega_3 & 1/Wt(\omega_3) \\ 1 & \cos\omega_4 & \cos 2\omega_4 & \cos 3\omega_4 & \cos 4\omega_4 & -1/Wt(\omega_4) \\ 1 & \cos\omega_5 & \cos 2\omega_5 & \cos 3\omega_5 & \cos 4\omega_5 & 1/Wt(\omega_5) \\ 1 & \cos\omega_6 & \cos 2\omega_6 & \cos 3\omega_6 & \cos 4\omega_6 & -1/Wt(\omega_6) \end{bmatrix} \begin{bmatrix} \mathrm{d}(0) \\ \mathrm{d}(1) \\ \mathrm{d}(2) \\ \mathrm{d}(3) \\ \mathrm{d}(4) \\ \mathrm{e_m} \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \end{bmatrix} \tag{7.5.17 a}$$

用矩阵符号简写为：

$$Pa*d=D \tag{7.5.17 b}$$

它和(7.5.8)式有些类似，差别在于：

（1）它的未知数 d 中，除了 $d(n)$外，还有一项 e_m。

（2） 方程数目与未知数数目相等。其表现是最左边的系数矩阵 Pa 是 6×6 的方阵。Pa=[P,WT'],其中 WT'是取决于加权系数 Wt 的单列向量，从(7.5.17)中可知 WT=(-1).^[0:L+1]./Wt。而在(7.5.8)式中，方程数大于未知数数，P 是一个长矩阵。这个方程的解也可用 MATLAB 语句 $d=Pa\backslash D$ 求得。

不过，上面提供的方法是在这些极值频点 ω_i 已知的前提下得到的。所以问题并未彻底解决，因为不知道这 $L+2$ 个极值点的频率 $\omega_i(i=1,2,\cdots,L+2)$，在什么准确位置。这个问题没有解析解，只能靠逐次迭代的方法来解决。从数学上，它的基础是 Remez 交换算法，而首先用它来解决滤波器设计的是 Parks-McClellan，所以这种算法就有两三种称呼。现就用例 7.5.1 的数据为例，用最小最大波动法来进行设计。

例 7.5.2：用最小最大波动法设计一个长度为 N=11 的 I 型线性相位 FIR 滤波器。要使它在 ω=[0,1] 之间的通带和[1.5,π]之间的阻带上以最小波动逼近理想的低通幅特性。设通带允许波动为阻带允许波动的 2 倍。

解：因为 N=11, L=5,最大波动极值点共有 $L+2$=7 个。但它们出现的位置我们并不知道。先任意在通带内设三个点[0, 0.5, 1]，阻带内设四个点[1.5, 2.0, 2.5, 3.14]。于是预期频率特性为

w=[0, 0.5, 1,1.5, 2,2.5,3.14];　D=[1, 1, 1, 0, 0, 0, 0];

加权系数为　　　　Wt=[0.5, 0.5, 0.5, 1, 1, 1, 1]

而方程(7.5.17)中的系数矩阵 Pa 可以按例 7.5.1 的方法先列写 P，再与加权系数有关的列向量 WT 拼接起来，因此可以得到如下的 MATLAB 程序 hc752。

```
N=11;L=floor((N-1)/2);                    % 滤波器及方程阶数
w0=[0, 0.5, 1, 1.5, 2.0  2.5  3.14];      % 初设频率向量
D=[1,1,1,0,0,0,0];                         % 预期幅频特性
P=cos(w0'*[0:L]);                          % 符幅特性谐波向量
Wt=[0.5,0.5,0.5,1,1,1,1];                  % 加权向量
WT=(-1).^[0:6]./Wt;                        % 求系数中的 WT 项
Pa=[P,WT' ];                               % 最小最大方程系数矩阵
d=Pa\ D'                                   % 求解最小最大方程，将 d 换为行向量
```

程序运行的结果为

```
d=    0.3924
      0.6083
      0.2060
     -0.1187
     -0.1934
     -0.0213
      0.0634
```

根据此处 d 向量的定义，它的最后一项 $d(7)=e_m$，只有前面的六项是式(7.5.19 a)中的 d，另外由于 MATLAB 中下标不允许取 0，程序中 d 的下标比公式中大 1。所以 MATLAB 给出的解 $d(1)$~$d(6)$,对应于公式中的 $d(0)$~$d(5)$。

同样，可以按表 7.5.1 中第一行末栏的公式，由 d 算出相应的滤波器系数 h，变成行向量，并进一步求出滤波器的频率特性，进行校核。在最小最大波动设计中，特别要校核初始设定的频率是不是在波动的极值点上，因此还要增加以下的语句：

```
h=[flipud(d(2:6)/2)',d(1),d(2:6)'/2]   % 按类型 I 的 FIR 滤波器由 d 求 h
[H,w]=freqz(h,1);                      % 由脉冲响应求频率特性
plot(w,abs(H)),hold on                 % 画出实际的幅频特性
H0=freqz(h,1,w0);e = abs(H0)-D         % 求给定频点上的幅特性，并求点上的误差
plot(w0,abs(H0),'.')                   % 画出设定频点上的幅特性
```

执行这几条语句后得到的结果为：

```
h=[ -0.0107  -0.0967  -0.0594  0.1030  0.3042  0.3924  0.3042  0.1030  -0.0594
   -0.0967   -0.0107]
e = -0.1267    0.1267    0.1267    0.0634    0.0634    0.0634    0.0634
```

所得滤波器的幅特性如图 7.5.4 所示。

可以看出，这个设计方程确实保证了误差按加权原则均匀分布。但是，问题在于所取的频点并不是最大波动的位置。所以要使设计满足要求，必须调整频点向量的数据。按曲线上目测的方法，除[0,1,1.5,3.14]四个边界点不动外，另外三个频点向极值点方向调整为 0.65，1.8,2.4，重新运行此程序，得到的结果

```
h = -0.0135  -0.0960  -0.0520  0.0955  0.2973  0.3916  0.2973  0.0955
    -0.0520  -0.0960  -0.0135
e = -0.1458  0.1458  -0.1458  0.0729  0.0729  0.0729  0.0729
```

相应的曲线见图 7.5.5。

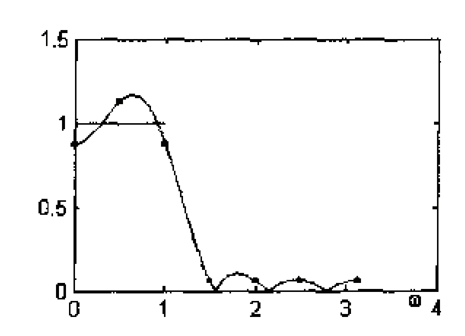


图 7.5.4　由初始给定频点设计出的幅特性

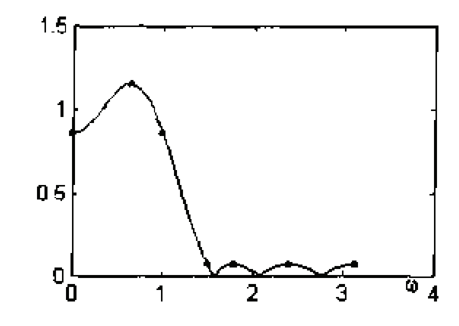


图 7.5.5　由初始修改后的频点设计出的幅特性

可以看出这些频点大体上都处在幅特性波动的极值点上，所以设计基本满足了最小最大波动的要求。这里我们是用目测试凑的方法来修正频点的位置，因而不能形成编写程序的规则，主要是说清最优化设计的思想和原理。完整的滤波器设计过程将在下一节结合 MATLAB 中已有子程序来讨论。

7.5.3　信号处理工具箱中的最优化等波动设计函数

要构成完整的最优等波动滤波器设计程序，除了以上的最小最大波动公式外，还有许多实际问题要考虑。

（1）滤波器长度 N(或阶数 $M=N-1$)如何确定？

（2）极值数目的确定。最优等波动滤波器的误差函数在 Ω 上有（$L+2$）或（$L+3$）个极值。大多数等波动滤波器有($L+2$)个极值。但是对于某些 ω_p、ω_s 的组合，可能得到有（$L+3$）个极值的滤波器。此处 L=floor(($N-1$)/2)。

（3）如何建立进行频率修正的算法？希望不要靠人们的主观观察，在程序中可以自动进行反复的迭代修正，直到达到要求的精度为止。

把最优等波动算法和上述的工程问题组合在一起，才能形成完整的设计算法，交替定理保证最大最小逼近问题的解存在并且惟一，但它并没有说明怎样得到这个解，既不知道阶数 N（或 L），也不知道极值频率 ω_i 和最大误差 δ。Parks 和 McClellan 在参考文献[17]中提供了利用 Remez 交换算法导出的迭代算法。它假设已知滤波器长度 N（或 L）和比率 δ_1/δ_2。如果按（7.5.12）选择了权函数，也正确地选择了阶数 N，并且设 $\delta=\delta_2$，就可得到解。显然 δ 和 N 是相关的，N 越大，δ 越小。滤波器技术指标中给出了 δ_1，δ_2，ω_p 和 ω_s，因此需要设定 N 的值。凯泽提出了一个简单的公式（7.5.20）来逼近 N。

$$N=\frac{-20\log_{10}\sqrt{\delta_1\delta_2}-13}{14.6\Delta f}+1,\quad \Delta f=\frac{\omega_s-\omega_p}{2\pi} \tag{7.5.18}$$

Parks_McClellan 算法首先猜设（$L+2$）个极值频率$\{\omega_i\}$，估计这些频率上的最大误差，接着按给定的各点，拟合一个 L 阶多项式(7.5.11)。然后在一个较细的网格上确定局部极大误差及其极值频率$\{\omega_i\}$，由这些新频率点拟合出一个新的 L 阶多项式，重复以上过程。一直进行至找到最优集$\{\omega_i\}$和全局最大误差 δ 为止。此迭代过程保证是收敛的。迭代结果得到多项式系数 $d(n)$，并最后算出滤波器脉冲响应 $h(n)$。

由于 N 是近似的，最大误差 δ 可能不等于 δ_2。如果出现这种情况，需要增加 N（若 $\delta>\delta_2$）或减小 N（若 $\delta<\delta_2$），再次用 remez 算法确定一个新的 δ。重复此过程至 $\delta\approx\delta_2$。这样就得到了等波动滤波器。

在 MATLAB 中，实现 Parks_McClellan 算法的函数为 remez，它最常用的句法为：

```
[h]=remez(M,f,A,weights,ftype)
```

它有几种调用形式：

[h]=remez(M,f,A)：设计一个 M 阶（滤波器的长度为 $N=M+1$）FIR 数字滤波器，它的频率响应在数组 f 和 A 中给定。长度为 N 的数组 h 是返回的滤波器系数（或脉冲响应）。数组 f 单位为 π，即 $0\leqslant f\leqslant 1$。其中包含各边缘频率，这些频率必须以递增次序排列，从 0 开始，到 1 结束。数组 A 为各指定频率上预期的幅度响应。f 和 A 的长度必须相等且为偶数（频率成对出现），形成依次的关心频带(通带或阻带)和不关心频带(过渡带)。每个关心频带中所用的权函数等于 1，这说明在每个频带中的容限（δ_i）是相同的。

[h]=remez(M,f,A,weights)：与上述情形相似，数组 weights 的长度是 f 的一半，它是每个关心频带的权函数。

[h]=remez(M,f,A,ftype)：与第一种情形类似，不过加了一个滤波器类型 ftype 作为输入变元。当变元是字符串“hilbert”或“differentiator”时，此函数相应地设计数字希尔伯特变换器或数字微分器。7.6 节中将介绍这两种类型的滤波器的设计问题

[h]=remez(M,f,A,weights,ftype)：与上一种情况类似，只是由数组 **weights** 指定每个频带的加权函数。

正如在讨论 Parks_McClellan 算法时指出的，为了使用程序 **remez**，首先必须根据（7.5.20）式假设滤波器的阶数，得到数组 h 中的滤波器系数后，还必须检查最小阻带衰减，并与给定的 A_s 比较，然后增加（或减小）滤波器的阶数。重复此过程直到得到期望的 A_s。现用下面的例子来说明此过程。

例 7.5.3：设计一个最优化等波动数字 FIR 低通滤波器，要求采用类型 I 的线性相位滤波器来实现。其技术指标同例 7.3.5，即 $\omega_p=0.2\pi, R_p=0.25\text{dB}$；$\omega_s=0.3\pi, A_s=50\ \text{dB}$

解：设计此滤波器的 MATLAB 程序 hc753 为：

```
wp = 0.2*pi; ws = 0.3*pi; Rp = 0.25; As = 50;   % 给定设计指标
delta1 = (10^(Rp/20)-1)/(10^(Rp/20)+1);          % 求通带绝对波动指标
delta2 = (1+delta1)*(10^(-As/20));                % 求阻带绝对波动指标
weights = [delta2/delta1 1];                      % remez 函数要求的加权向量
deltaf = (ws-wp)/(2*pi);                          % 估算N需要的过渡带宽度
N = ceil((-20*log10(sqrt(delta1*delta2)) -13)/(14.6*deltaf)+1) % 估算N
N=N+mod(N-1,2);                                   % 对于类型I滤波器，N必须取奇数
f = [0 wp/pi ws/pi 1];                            % 函数要求的频率向量
m = [1 1 0 0];                                    % remez 函数要求的理想幅特性向量
h = remez(N-1,f,m,weights);                       % 用最优等波动法设计滤波器系数
[db,mag,pha,grd,w] = myfreqz(h,[1]);              % 计算滤波器频率特性
dw = 2*pi/1000; wsi=ws/dw+1; wpi = wp/dw;         % 求边缘频率下标
Asd = -max(db(wsi:1:501));                        % 求阻带波动
```

检查 Asd，若达不到要求，增加滤波器长度 $N = N+2$，（保持奇数）,若超过要求，则取 $N=N-2$，重复执行程序中有下画线的三条语句。

此程序第一次运行后得到 $N=43$ 及 Asd = 47.8562。未达到给定指标。

令 $N=N+2=45$，再执行这三条设计和检验语句，得到 Asd = 48.8632，仍未达到指标。

再令 $N=N+2=47$，执行这三条语句后，得到 Asd = 51.0896

由于算得的阻带衰减超过给定的 A_s，迭代过程终止，最后得到的 N 值为 47。这个值比采用窗设计技术（凯泽窗的 $N=61$）或频率样本技术（$N=60$）要小得多。

在程序的最后加上几条绘图语句，可以得到形象的结果。

```
[A,w1]=amplres(h);                       % 求符幅特性
subplot(2,2,1),stem([1:N],h,'.')         % 画滤波器脉冲响应
subplot(2,2,2),plot(w,db)                % 画分贝幅特性
subplot(2,2,3),plot(w,A)                 % 画符幅特性曲线
subplot(2,2,4),plot(w(1:wpi),A(1:wpi)-1,w(wsi:501),A(wsi:501))% 画误差波动曲线
```

得出的曲线见图 7.5.6。(a)子图给出此滤波器的时域响应曲线，(b)、(c)子图给出其频域中的幅特性和符幅特性，(d)子图给出其在通带及阻带中的误差函数，可以说明等波动特性和加权函数的作用。

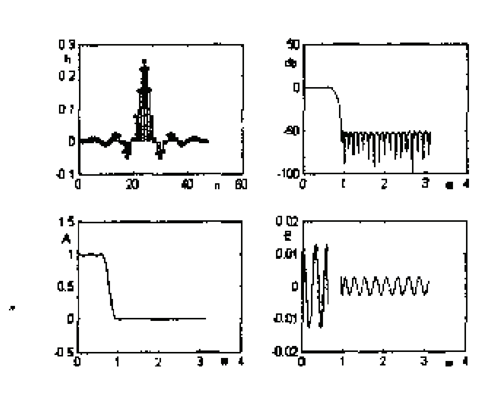


图 7.5.6　例 7.5.3 设计的最优化等波动滤波器的脉冲响应、幅特性和误差特性

例 7.5.4：设计一个最优化等波动数字 FIR 低通滤波器，其技术指标为

$\omega_{1s}=0.2\pi$,

$\omega_{1p}=0.35\pi$,　$R_p=0.25$dB

$\omega_{2p}=0.65\pi$,

$\omega_{2s}=0.8\pi$,　　$A_s=50$ dB

解：设计此滤波器的 MATLAB 程序 hc754 为：

```
ws1 = 0.2*pi; wp1 = 0.35*pi; wp2 = 0.65*pi; ws2 = 0.8*pi;  % 给定设计指标
Rp = 1.0; As = 60;
delta1 = (10^(Rp/20)-1)/(10^(Rp/20)+1);          % 求通带绝对波动指标
delta2 = (1+delta1)*(10^(-As/20));               %  求阻带绝对波动指标
weights = [1 delta2/delta1 1];                   % remez 函数要求的加权向量
delta_f =min((ws2-wp2)/(2*pi), (wp1-ws1)/(2*pi));   % 估算 N 需要的过渡带宽度
N = ceil((-20*log10(sqrt(delta1*delta2))-13)/(14.6*delta_f)+1) % 估算 N
f = [0 ws1/pi wp1/pi wp2/pi ws2/pi 1];           %  函数要求的频率向量
m = [0 0 1 1 0 0];                               % remez 函数要求的理想幅特性向量
h = remez(N-1,f,m,weights);                      % 用最优等波动法设计滤波器系数
[db,mag,pha,grd,w] = myfreqz(h,[1]);             % 计算滤波器频率特性
dw=2*pi/1000;                                    % 求边缘频率下标
ws1i=floor(ws1/dw)+1; wp1i = floor(wp1/dw)+1;
ws2i=floor(ws2/dw)+1; wp2i = floor(wp2/dw)+1;
Asd = -max(db(1:1:ws1i))                         % 求阻带波动。
```

检查 Asd，若达不到要求，增加滤波器长度 $N=N+1$，若超过要求，则取 $N=N-1$，重复执行程序中有下画线的三条语句。

此程序第一次运行后得到 $N=28$ 及 Asd = 56.5923，未达到给定指标。

令 $N=N+1=29$，再执行这三条设计和检验语句，得到 Asd = Asd = 61.2818，达到了指标。

由于算得的阻带衰减超过给定的 A_s，迭代过程终止，最后得到的 N 最优值为 29，在程序的最后加上几条绘图语句，可以得到形象的结果。

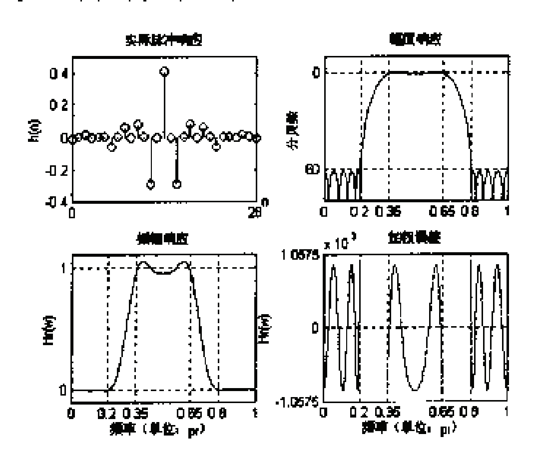


图 7.5.7　例 7.5.4 设计的最优化等波动带通滤波器的脉冲响应、幅特性和误差特性

得出的曲线见图 7.5.7，(a)子图给出此滤波器的时域响应曲线，(b)、(c)子图给出其频域中的幅特性和符幅特性，(d)子图给出其在通带及阻带中的误差函数，可以说明等波动特性和加权函数的作用。

7.6　FIR 滤波器设计的一些深入问题

7.6.1　关于四种线性相位滤波器类型的应用范围

到现在为止，我们所设计的滤波器都属于选频型的滤波器，包括低通、高通、带通和带阻。它们的预期幅频特性由逐段为常数的直线组成，并对零频率成偶对称。可以证明，这类滤波器的脉冲响应必然具有偶

对称的特性，也就是说，它们必然属于第 I 类或第 II 类线性相位滤波器。

N 为奇数的类型 I 线性相位滤波器具有最广泛的用途。因为它的符幅特性在 $\omega=0$ 和 $\omega=\pi$ 处的取值没有任何限制，所以可以用来做低通、高通、带通和带阻四种滤波器中的任何一种。从表 7.2.1 和公式（7.2.3）可知，N 为偶数的类型 II 滤波器的幅特性在 $\omega=\pi$ 处必定为零。这就限制了它的应用范围，它不能用于高通和带阻滤波器，因为这些滤波器在 $\omega=\pi$ 的高频区幅特性为 1。

理想低通滤波器的脉冲响应已经在第 5.5.1 节和第 7.3.1 节中详细讨论过。这里把其他三种理想滤波器也讨论一下。理想高通滤波器的脉冲响应为

$$h_{HP}(n)=\begin{cases}1-\omega_c/\pi & n=0\\ -\sin\omega_c n/\pi n & |n|>0\end{cases} \tag{7.6.1}$$

这个序列是偶对称的。由于它在 $n=0$ 的对称中心处必须有样本点，因此，在加窗和右移后的样本总数必须是奇数，所以它必定属于类型 I 线性相位滤波器。从表 7.2.1 可以看出，类型 II 的线性相位滤波器在 $\omega=\pi$ 处的幅特性必须为零，而这与高通滤波器的要求相矛盾。所以高通滤波器只能用 I 型 FIR 滤波器来实现。

同样可以导出，理想带通滤波器的脉冲响应为

$$h_{BP}(n)=\frac{\sin(\omega_{c2}n)}{n\pi}-\frac{\sin(\omega_{c1}n)}{n\pi} \qquad |n|\geqslant 0 \tag{7.6.2}$$

理想带阻滤波器的脉冲响应为

$$h_{BS}(n)=\begin{cases}1-(\omega_{c2}-\omega_{c1})/\pi & n=0\\ (\sin\omega_{c1}n-\sin\omega_{c2}n)/\pi n & |n|>0\end{cases} \tag{7.6.3}$$

由于同样的理由，要求高频能通过的带阻滤波器也不能采用 II 型滤波器，它也只能用 I 型滤波器来实现。而带通滤波器则可以用类型 I 或 II 的线性相位滤波器实现。

除了以上的低通、高通、带通和带阻滤波器外，实践中还有多波段滤波器，在 0～π 范围内可能有多个通带和阻带，而且其预期幅特性也不一定只是 0 和 1，可以有其他中间值。这些情况下的设计可以用信号处理工具箱中的 fir2 函数直接进行设计。其结果也必定是偶对称的线性相位滤波器，凡是有高通要求的就必须用类型 I。

类型 III 和 IV 的 FIR 滤波器的系数向量是反对称的，系数的累加平均值为零，意味着它的直流分量为零，从其幅频特性上可以看出，它在零频率处幅特性为零，所以它拒绝一切直流的分量通过。III 型的还拒绝高频分量通过，所以这两类滤波器都不适合作为选频之用。

类型 III 和 IV 的 FIR 滤波器有什么用处呢？注意它们在零频率处的相特性都是 π/2。这个特性恰好适合某些信号变换的要求，这就是数字微分器和希尔波特变换器。这两种用途所要求的特性目前可能还不容易为读者所完全理解，但为了对四种 FIR 滤波器有一个全面的概念，而且知道所介绍的三种 FIR 滤波器设计方法和相应的 MATLAB 设计函数可以同样用在这两类变换器上，本节将对其做一扼要的介绍。

7.6.2 数字微分器的设计

严格地说，微分器只是为连续系统定义的。如果对 $x(t)$ 的拉普拉斯变换是 $X(s)$，则对 $x(t)$ 的导数的拉普拉斯变换是：

$$L\left[\frac{\mathrm{d}x(t)}{\mathrm{d}t}\right]=sX(s)=H_a(s)X(s) \tag{7.6.4}$$

所以连续时间微分器的传递函数是 s，因而它的频率响应则是 $H_a(\mathrm{j}\Omega)=\mathrm{j}\Omega,(-\infty<\Omega,\infty)$。可以把连续微分器按图 1.2.2 那样，经过采样，变成数字信号，在数字域进行微分后再恢复为模拟量，这样就可以用数字方法来近似实现连续信号的微分。因为信号一旦被数字化以后，采样周期 T 在数字处理中就不起作用了。可以假设 $T=1$ 而不失一般性。对于一个长度为 N 的数字微分器，其理想频率响应为

$$H_d(\omega) = j\omega e^{-j0.5(N-1)\omega} = j\omega e^{-j\tau\omega} \tag{7.6.5}$$

其中的相位特性 $-\tau\omega=-0.5(N-1)\omega$ 是因果性的要求，即保证其脉冲序列是右序列决定的。

从时域看，最简单的微分器的差分方程是

$$y(n)=x(n)-x(n-1) \tag{7.6.6}$$

它的系数[1,-1]是反对称的，长度是偶数。这是一个类型 IV 的一阶 FIR 滤波器。 做 z 变换，得到

$$Y(z)=X(z)-z^{-1}X(z)=(1-z^{-1})X(z)$$

因而

$$H_1(z)=\frac{Y(z)}{X(z)}=1-z^{-1}=\frac{z-1}{z} \tag{7.6.7}$$

它的频率特性为

$$H_1(e^{j\omega})=1-e^{-j\omega}=e^{-j0.5\omega}(e^{j0.5\omega}-e^{-j0.5\omega})=j2\sin 0.5\omega\times e^{-j0.5\omega} \tag{7.6.8}$$

(7.6.5)式是理想的微分器的频率特性，而(7.6.8)式是它的一阶近似。其相位特性部分(包括相位为 $\pi/2$ 的乘子 j)相同；而其符幅特性部分则不相同，理想情况为 $A(\omega)=\omega$,一阶近似为 $A_1(\omega)=2\sin(0.5\omega)$,在 ω 很小时，$\sin(x)\approx x$，有 $A_1(\omega)=A(\omega)\approx\omega$。所以对于缓慢变化的连续信号，可以用简单的一阶数字微分器来计算它的导数。如果输入信号变化的带宽较大，那就要尽量精确的实现(7.6.5)式，要用阶次比较高的宽带微分器。

计算宽带微分器频率特件(7.6.5)式的 IDFT，可以得到它的理想脉冲响应。设 $\tau=(N-1)/2$,可得

$$h_d(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi} j\omega e^{j(n-\tau)\omega}d\omega=\frac{1}{2\pi(n-\tau)}\int_{-\pi}^{\pi}\omega\, de^{j(n-\tau)\omega}$$

并采用分部积分，得到

$$h_d(n)=\frac{\cos[(n-\tau)\pi]}{n-\tau}-\frac{\sin[(n-\tau)\pi]}{\pi(n-\tau)^2} \tag{7.6.9}$$

当 $N=2\tau+1$ 是奇数时，τ 是整数，对于所有整数 n，$\sin[(n-\tau)\pi]=0$。 (7.6.9)中的后一项为零，只剩下第一项余弦项。因此，有

$$h_d(n)=\begin{cases}\dfrac{\cos[(n-\tau)\pi]}{n-\tau} & n\neq\tau\\ 0 & n=\tau\end{cases} \tag{7.6.10}$$

当 $N=2\tau$ 是偶数时，τ 是整数/2，对于所有整数 n，$\cos[(n-\tau)\pi]=0$。(7.6.9)式中的前一项为零，只剩下第二项正弦项。因此，有

$$h_d(n)=-\frac{\sin[(n-\tau)\pi]}{\pi(n-\tau)^2} \tag{7.6.11}$$

公式(7.6.10)和 (7.6.11)都满足 $h_d(n)=-h_d(N-n-1)$，即都是反对称的无穷序列。如果用长度为 N 的窗函数 Wd (n)对它进行截断，就得到可以实现的实际的微分器，其脉冲响应（也就是微分器的系数向量）为

$$h(n)=h_d(n)\mathrm{Wd}(n) \tag{7.6.12}$$

这就得到了用窗函数法设计的 FIR 微分器（滤波器）系数向量。

实际上，在类型 III 和 IV 中，只有 IV 型 FIR 滤波器适合做微分器。因为微分器的幅特性随 ω 的增加线性地增长，在 $\omega=\pi$ 处应达到最大。而从表 7.2.1 看出，III 型 FIR 滤波器在 $\omega=\pi$ 处的幅特性必须为零。所以只有用 N 为偶数的 IV 型 FIR 滤波器，才能用较低的阶次获得较好的微分效果。

例 7.6.1：用矩形窗和哈明窗分别设计一个长度为 $N=6$ 的数字微分器。

解：此题的 MATLAB 程序 hc761 非常简单，语句如下：

```
N=6; tau=(N-1)/2; n=[0:N-1]+eps;              % 设定微分器长度
hd =-sin((n-tau).*pi)./(pi.*(n-tau).^2);      % 计算其矩形窗截断脉冲响应
hh=hd.*hamming(N)';                           % 加哈明窗后的系数向量
[Hd,wd]=freqz(hd,1);                          % 计算矩形窗截断微分器频率响应
```

```
[Hh,wh]=freqz(hh,1);                                    % 计算哈明窗截断微分器频率响应
plot(wd,abs(Hd),wh,abs(Hh),':',wd,abs(wd),'--'),hold on  %绘制幅特性
legend('矩形窗','哈明窗','理想'),grid on
```

执行这个程序后得到的滤波器系数向量如下：

```
hd =   0.0509  -0.1415   1.2732  -1.2732   0.1415  -0.0509
hh =   0.0041  -0.0563   1.1614  -1.1614   0.0563  -0.0041
```

两种窗函数设计的幅频特性曲线及理想幅频特性（直线）均示于图 7.6.1 中。

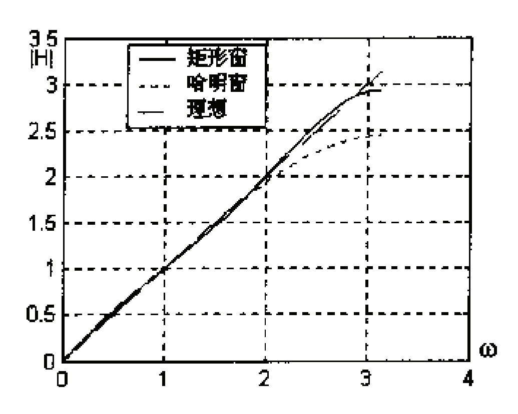


图 7.6.1　例 7.6.1 的五阶数字微分器的幅频特性

数字微分器也同样可以用频率样本法和最优化方法设计。在 MATLAB 信号处理工具箱函数中，最小二乘设计函数 firls 和最优等波动设计函数 remez 都有最后的一个任选输入变元，如果键入‘differentiator’,它就进行数字微分器的设计。例如：

```
b=firls(N-1,f,A,'differentiator')      % 进行最小二乘设计
b1=remez(N-1,f,A,'differentiator')     % 进行最优等波动设计
```

其中：N 为滤波器长度；N-1 为微分器阶次；f 为成对的频率向量,单位为 π，取值范围为 $0 \leq f \leq 1$；A 为这些频率上的幅度向量，f 和 A 规定了预期频率特性。按上例的情况，预期频率特性简单地由一段斜线表示，因此只要首尾两点就能表述，f 和 A 都只需有两个元素。将数据输入如下：

```
f = [0, 1], A=[0, pi]; N=6;
b = firls(N-1, f, A, 'differentiator')
b1 = remez(N-1, f, A, 'differentiator')
```

得到的滤波器系数分别为

```
b =    0.0306  -0.1359   1.2699  -1.2699   0.1359  -0.0306
b1 =   0.0508  -0.1631   1.2829  -1.2829   0.1631  -0.0508
```

最小二乘设计的结果和矩形窗函数法相近，这是可以预想到的。有关这些函数的更多的信息，可用 help 语句获得。

7.6.3　Hilbert 变换器的设计

理想希尔波特(Hilbert)变换器的特性可表示为

$$H_d(e^{j\omega}) = \begin{cases} -j & 0 < \omega < \pi \\ j & -\pi < \omega < 0 \end{cases} \tag{7.6.13}$$

它的幅特性在全频段上都是 1，所以相当于一个全通滤波器。它对 $\omega<0$ 的频段引入 -90° 相移，而对 $\omega<0$ 的频段引入 90° 相移，所以也称它为 90° 相移器。

先介绍希尔波特变换器的用处。考虑一个限带离散时间信号 $x(n)$，其频谱函数为 $X(\mathrm{e}^{\mathrm{j}\omega})$，带宽为 W，幅特性如图 7.6.2(a)所示。为了进行传送，它被调制到 ω_0 的载频上，成为 $x(n)\cos\omega_0 n$。这个调制信号的频谱是两个相乘信号频谱的卷积。由表 3.3.2 得知，$\cos\omega_0 n$ 在奈奎斯特频段内的频谱是位于 $\pm\omega_0$ 处的两个 δ 函数，它与 $X(\mathrm{e}^{\mathrm{j}\omega})$ 卷积的结果是把中心频率为 0 的 $X(\mathrm{e}^{\mathrm{j}\omega})$ 分成两半，分别移到中心频率为 $\pm\omega_0$ 处，如图 7.6.2 (b)所示。这样调制后的信号带宽就成为 $2W$。如果能把信号带宽减小一半，那么信道中传输的信号量就可以加倍。这在工程中和经济上都有很大的价值。

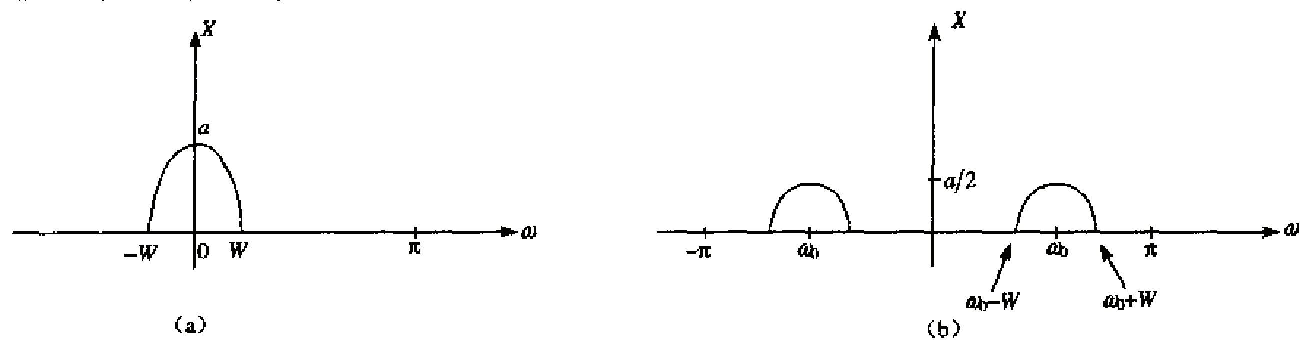


图 7.6.2　信号频谱(a)和调制后信号频谱(b)

因为信号的频谱是复数，可以分解为实部 X_{re} 和虚部 X_{im}，

$$X(\mathrm{e}^{\mathrm{j}\omega}) = X_{\mathrm{re}}(\mathrm{e}^{\mathrm{j}\omega}) + \mathrm{j}X_{\mathrm{im}}(\mathrm{e}^{\mathrm{j}\omega})$$

如果 $x(n)$ 是实信号，则它的频谱是共轭对称的。即 $X_{\mathrm{re}}(-\omega) = X_{\mathrm{re}}(\omega)$，及 $X_{\mathrm{im}}(-\omega) = -X_{\mathrm{im}}(\omega)$。

如果让信号 $x(n)$ 通过(7.6.13)式所表示的希尔波特变换器，则它的输出信号 $y(n)$ 的频谱为

$$Y(\mathrm{e}^{\mathrm{j}\omega}) = H_{\mathrm{d}}(\mathrm{e}^{\mathrm{j}\omega})X(\mathrm{e}^{\mathrm{j}\omega}) = \begin{cases} -\mathrm{j}X_{\mathrm{re}}(\mathrm{e}^{\mathrm{j}\omega}) + X_{\mathrm{im}}(\mathrm{e}^{\mathrm{j}\omega}) & 0<\omega<\pi \\ \mathrm{j}X_{\mathrm{re}}(\mathrm{e}^{\mathrm{j}\omega}) - X_{\mathrm{im}}(\mathrm{e}^{\mathrm{j}\omega}) & -\pi<\omega<0 \end{cases} \tag{7.6.14}$$

然后把 $x(n)$ 和 $y(n)$ 组成复数信号 $z(n)=x(n)+\mathrm{j}y(n)$，则 $z(n)$ 的频谱函数为式（7.6.15）所示。

$$Z(\mathrm{e}^{\mathrm{j}\omega}) = X(\mathrm{e}^{\mathrm{j}\omega}) + \mathrm{j}Y(\mathrm{e}^{\mathrm{j}\omega}) = \begin{cases} 2X(\mathrm{e}^{\mathrm{j}\omega}) & 0<\omega<\pi \\ 0 & -\pi<\omega<0 \end{cases} \tag{7.6.15}$$

它在 $\omega<0$ 的频段内的频谱为零，只保留了 $\omega>0$ 的频段内的频谱，所以其频带宽度减小了一半。由于频谱的共轭对称性，它也保留了原来频谱中的所有信息，可以在接收端完全恢复原来的信号内容。这是希尔波特变换器（滤波器）的重要用途。它的结构框图见图 7.6.3。

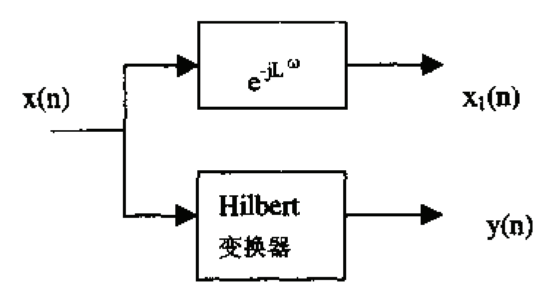


图 7.6.3　希尔波特变换器用于压缩频带

同样由于非因果和非绝对可加的问题，理想希尔波特变换器是无法实现的。下面来讨论如何用 $N-1$ 阶的线性相位 FIR 滤波器来近似实现希尔波特变换器的问题。先按因果性的要求，把脉冲序列移后 $-0.5(N-1)$ 拍，(7.6.13)式就移相 $-0.5(N-1)\omega$，得到

$$H_d(e^{j\omega})=\begin{cases}-je^{-j0.5(N-1)\omega} & 0<\omega<\pi \\ je^{j0.5(N-1)\omega} & -\pi<\omega<0\end{cases} \tag{7.6.16}$$

其中根据 N 是奇数或偶数，0.5(N-1)可能是整数或非整数。如果它不是整数，在按图 7.6.3 把虚实信号叠合为 $x(n)+jy(n)$时，会造成时间匹配的困难，所以实际上只取 N 为奇数。

现在计算(7.6.14)式的 IDFT，其结果应该就是此滤波器的系数向量

$$h_d(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(e^{j\omega})e^{jn\omega}d\omega$$

用分部积分法，引入 $\tau=(N-1)/2$，最后得到

$$h_d(n)=\frac{2\sin^2[(n-\tau)\pi/2]}{(n-\tau)\pi} \tag{7.6.17}$$

很容易证明，$h_d(n-\tau)=0$,及 $h_d(n)=-h_d(N-1-n)$ ，所以系数序列是反对称的。而序列的长度 N 为奇数，故它是类型 III 的 FIR 线性相位滤波器。

由于(7.6.15)式给出的是用矩形窗截止并移位后的脉冲响应，它也就是滤波器的系数向量，所以用窗函数法设计希尔波特变换器变得非常容易。只要把 $h_d(n)$ 与窗函数相乘即可。下面的例子显示设计的过程和特点。

例 7.6.2：用矩形窗和汉宁窗设计一个十阶的希尔波特变换器。

解：可以直接写出设计希尔波特变换器的程序。其设计语句很简单，而检验语句和以往有些不同。因为希尔波特变换器在正负频率区间的相位特性不同，要检验这一点，就必须计算它在全频域的频率特性，因此在 freqz 函数中加变元 'whole'；另外，MATLAB 的相特性通常只取主值-π～π。在本题中，要看出希尔波特变换器构成的相位突变，就不希望出现因取相角主值带来的突变。因此用了 unwrap(angle(H))代替通常的 angle(H)，这样，就取全相角而不是它的主值了。本例的 MATLAB 程序 hc762 如下。

```
N=input('滤波器长度=(输入奇数) ');            % 给出滤波器长度，它等于阶数加一
tau=(N-1)/2;n=[0:2*tau]+1e-10;               % 给出 n 数组，加微小偏移
hd=2*(sin((n-tau)*pi/2)).^2./(n-tau)/pi;     % 矩形窗截断的系数向量
hhn=hd.*hann(N)';                             % 汉宁窗截断的系数向量
[H,w]=freqz(hd,1,'whole');                    % 求滤波器全频域频率响应
[Hhn,w]=freqz(hhn,1,'whole');
```

运行此程序，输入 N=11,得到

```
hd = -0.1273  0 -0.2122 0  -0.6366  0  0.6366  0  0.2122  0  0.1273
hhn = 0   0 -0.0733  0  -0.5758  0  0.5758    0    0.0733  0     0
```

程序运行后得到的曲线如图 7.6.4 所示。

计算结果为图 7.6.4(a)中画出的变换器的系数向量。除了证明它是反对称的 III 型 FIR 滤波器外，发现系数 hd 的另一特点，那就是从对称中心 $n=\tau$ 处向两边延伸，每间隔一个位置，系数就为零。本例名义上系数长度为 11，实际只有六个系数，其余五个系数为零。由于反对称，六个系数只有三个是独立可变的。如果我们取 N=13,其结果不过是在两端各加一个零系数，与 N=11 没有什么差别。因此，希尔波特变换器的长度应该取为 4I-1(I=1,2,3⋯)，即 N 应取 3,7,11,15⋯。

理想的希尔波特变换器应该具有在全频段幅度恒为 1 的幅频特性。子图(b)画出了实际变换器的幅特性。因为 III 型滤波器在 ω=0 和 π 处的幅特性必须为 0。在这两个突变的边缘上，用矩形窗就会因吉布斯效应而出现波动。子图(d)说明用汉宁窗可以大大地抑制波动。从子图(c)可以看出在正负频率的交界处出现相角的突跳，按$-0.5\tau\omega$ 的人为延迟（虚线）为基准，在正频率区（此处为 $0<\omega<\pi$ 区）相移为$-\pi/2$,而在负频率区（此处为 $\pi<\omega<2\pi$ 区）相移为 $\pi/2$。符合希尔波特变换器的要求。

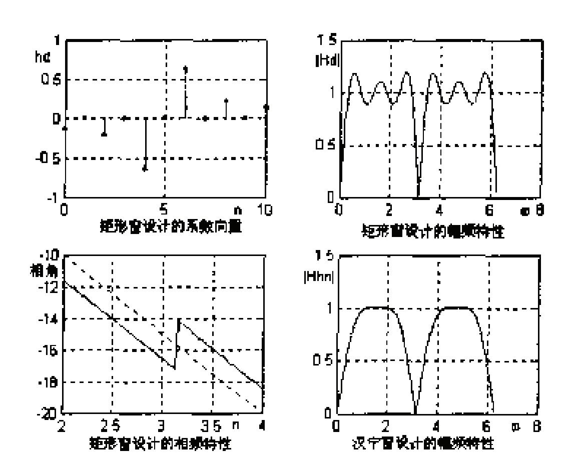


图 7.6.4　N=11 的希尔波特变换器设计的结果，左为系数向量，右为幅频特性

希尔波特变换器也同样可以用频率样本法和最优化方法设计。在 MATLAB 信号处理工具箱函数中，最小二乘设计函数 firls 和最优等波动设计函数 remez 都有最后的一个任选输入变元，如果键入'Hilbert',它就进行希尔波特变换器的设计。例如：

```
b=firls(N-1, f, A, 'Hilbert')          % 进行最小二乘设计
b1=remez(N-1, f, A, 'Hilbert')         % 进行最优等波动设计
```

其中：N 为滤波器长度；N-1 为希尔波特变换器阶次；f 为成对的频率向量；A 为这些频率上的幅度向量。f 和 A 规定了预期频率特性。按上例的情况，预期频率特性简单地由一段斜线表示，因此只要首尾两点就能表述，f 和 A 都只需有两个元素，将数据输入如下：

```
f=[0,1],A=[1,1]; N=11;
b=firls(N-1,f,A,'Hilbert')
b1=remez(N-1,f,A,'Hilbert')
```

得到的滤波器系数分别为

```
b = -0.1273   0  -0.2122  0  -0.6366  0   0.6366  0  0.2122   0  0.1273
b1 = -0.4453  0  -0.2227  0  -0.6392  0   0.6392  0  0.2227  0   0.4453
```

最小二乘设计的结果和矩形窗函数法相近，这是可以预想到的。有关这些函数的更多的信息，可用 help 语句获得。

复习思考题

7.1 为什么理想低通滤波器是不能实现的？

7.2 滤波器是用幅度特性及相位特性来确定的。为什么在设计时经常只规定幅度特性？

7.3 相位失真的影响如何体现？为什么要求线性相位？它的失真影响又如何体现？

7.4 为什么在设计 FIR 滤波器时稳定性不成问题？

7.5 为什么要引进符幅特性 $A(\omega)$的概念？它与幅特性有何关系？怎样求符幅特性？

7.6 FIR 滤波器的系数为什么必须对称？对称的数学表述是什么？

7.7 何谓截断？什么是它所带来的吉布斯现象？增加长度 N 能消除这个现象吗？消除它的办法有哪些？要付出什么代价？

7.8 什么是三角窗、布拉克曼窗和汉明窗？在使用这些窗时，为什么可以减少波动？

7.9 频率样本法在选定频率样本的幅特性和相特性时，应遵循哪些规则？

7.10 连续最小二乘法的目标函数是什么？此目标函数最优化的结果是什么？

7.11 离散最小二乘法的目标函数是什么？其方程组的数目和未知数数目各由什么条件决定？怎样才能保证方程有解。

7.12 极小化最大波动的意思是什么？它有什么样的优点？

7.13 在什么意义下等波纹滤波器是最佳的？

7.14 为什么 I 型线性相位滤波器的用途最为广泛？II 型滤波器用做选频滤波器受到什么限制？

7.15 为什么 III 型和 IV 型滤波器不宜于用做选频滤波器？它们的用途是什么？

习题

7.1 设一个四阶 FIR 滤波器的系数为：$h(0)=h(4)=2, h(1)=h(3)=-1, h(2)=3$，试导出它的频率特性表示式和符幅特性表示式。

7.2 用矩形窗求出一个四阶 FIR 滤波器的系数，使它逼近理想低通滤波器：在$|\omega|\leqslant 1$ 处，$H(\omega)=\mathrm{e}^{-\mathrm{j}2.5\omega}$，在$|\omega|\geqslant 1$ 处，$|H(\omega)|=0$。如果用 Hamming 窗，系数又如何？

7.3 设低通滤波器的单位脉冲响应为 $h(n)$，如果另一个滤波器的单位脉冲响应为 $h_1(n)$，它与 $h(n)$ 的关系是 $h_1(n)=(-1)^n h(n)$，试证明 $h_1(n)$ 代表一个高通滤波器。

7.4 设低通滤波器的单位脉冲响应与传递函数分别为 $h(n)$ 和 $H(\mathrm{e}^{\mathrm{j}\omega})$，截止频率为 ω_c。如果另一个滤波器的单位脉冲响应为 $h_2(n)$，它与 $h(n)$的关系是 $h_2(n)=2h(n)\cos\omega_c n$，试证明 $h_2(n)$代表一个带通滤波器。

7.5 II 型线性相位滤波器的系数特征为 $h(n)=h(N-1-n)$，　$0\leqslant n\leqslant N-1$，$N$ 为偶数。试证明它的符幅特性 $A(\omega)$可以由下式表征。

$$A(\omega)=\sum_{n=0}^{L}d(n)\cos[\omega(n-1/2)]$$

并求出其中的 $d(n)$与 $h(n)$的关系式和 L 与 N 的关系式。

7.6 设通带边缘频率 $\omega_p=\pi/4$ rad，$N=21$,分别利用矩形窗、哈明窗、汉宁窗和布莱克曼窗设计 FIR 滤波器，求出它们的符幅特性并进行比较。

7.7 求出习题 7.6 中各个滤波器的累积幅特性并进行比较。

7.8 如果 $H(z)$在 $z_1=r\mathrm{e}^{\mathrm{j}\theta}, z_2=\frac{1}{r}\mathrm{e}^{-\mathrm{j}\theta}, z_3=r\mathrm{e}^{-\mathrm{j}\theta}, z_4=\frac{1}{r}\mathrm{e}^{\mathrm{j}\theta}$ 四个位置有零点，证明 $H(z)$ 是一个线性相位 FIR 滤波器。

7.9 用汉宁窗设计技术设计一个带阻滤波器，技术指标为：

低阻带边缘：　0.4π

高阻带边缘：　0.6π　　$A_s=40$ dB

低通带边缘：　0.3π

高通带边缘：　0.7π　　$R_p=0.5$ dB

画出设计的滤波器的脉冲响应和幅度响应（dB 值）。

7.10 用哈明窗设计技术设计一个带通滤波器，技术指标为：

低阻带边缘：　0.3π

高阻带边缘：　0.6π　　$A_s=50$ dB

低通带边缘：　0.4π　　$R_p=0.5$dB

高通带边缘：　0.5π

画出设计的滤波器的脉冲响应和幅度响应（dB 值）。

7.11 用凯泽窗设计技术设计一个高通滤波器，技术指标为：

阻带边缘：　0.4π　$A_s = 60\ \text{dB}$

通带边缘：　0.6π　$R_p = 0.5\ \text{dB}$

画出设计的滤波器的脉冲响应和幅度响应（dB 值）。

7.12 设预期频率特性具有直线形式的过渡带和线性相位 $H_d(\omega) = D(\omega)e^{-j\omega(N-1)/2}$，即其预期符幅特性 $D(\omega)$可表示为下式：

$$D(\omega) = \begin{cases} 1 & 0 \leqslant \omega \leqslant \omega_p \\ (\omega_s - \omega)/(\omega_s - \omega_p) & \omega_p < \omega < \omega_s \\ 0 & \omega_s \leqslant \omega \leqslant \pi \end{cases}$$

设 $\omega_p=1, \omega_s=1.5$，求出它的 IDTFT。用 N=9，21 和 61，对它截断，得到 N-1 阶 FIR 滤波器。分别求出它们的幅特性，观察有无吉布斯效应，并作讨论。

7.13 对题 7.12 中的预期频率特性，用频率样本法设计一个六阶和一个七阶 FIR 滤波器，检验它们是否具有线性相位。

7.14 对题 7.12 中的预期频率特性，用离散最小二乘法设计一个六阶和一个七阶 FIR 滤波器，使滤波器在 $\omega=0, \omega=\omega_p=1, \omega=\omega_s=1.5$ 三个频点上满足最小二乘目标函数。

7.15 用频率样本法设计题 7.10 中的带通滤波器，适当选择滤波器阶数，使得过渡带中有两个样本，设计过程中使用这些样本的最优值。

7.16 用频率样本法设计习题 7.11 中的高通滤波器，选择适当的滤波器阶数，使得过渡带中有两个样本，设计过程中使用这些样本的最优值。

7.17 对题 7.12 中的预期频率特性，用离散最小二乘法设计一个八阶 FIR 滤波器，使滤波器在 13 个频点上满足最小二乘目标函数。其中五个频点在通带中均匀分布，七个频点在阻带中均匀分布，还有一个频点在过渡带的中点。

7.18 用 MATLAB 中的 firls 函数解题 7.11，画出其频率特性与题 7.16 相比较。

7.19 设计一个窄通带滤波器，通过的中心频率为 $\omega_0 = 0.5\pi$，其带宽应不大于 0.1π。

(a) 利用频率采样技术，合理选择滤波器长度 N，使得过渡带中有一个样本，设计中使用过渡带样本的最优值，并画出频率采样结构。

(b) 利用凯泽窗技术，使阻带衰减与上面的频率采样设计中的结果相同，确定脉冲响应 $h(n)$，并画出线性相位结构。

(c) 就上述两种滤波器设计的实现和经济性进行比较。

7.20 理想带通滤波器的幅频特性为：

$$D(\omega) = \begin{cases} 0 & 0 < |\omega| \leqslant \pi/3 \\ 1 & \pi/3 < |\omega| \leqslant 2\pi/3 \\ 0 & 2\pi/3 < |\omega| \leqslant \pi \end{cases}$$

(a) 根据 Remez 算法，确定有 25 个抽头的滤波器的系数，它的阻带衰减为 50 dB，设计的滤波器应该尽可能地做到具有最小的过渡带宽。

(b) 计算并画出滤波器的符幅特性。

7.21 用信号处理工具箱 fir1 函数直接设计阶数最低的线性相位 FIR 滤波器，满足题 7.9、

题 7.10 和题 7.11 中的指标要求，并和该题的原来题解作比较。

7.22 设计一个八阶离散最小二乘 FIR 滤波器来逼近具有以下幅特性的理想高通滤波器

$$D(j\omega)=\begin{cases}0 & 0\leqslant|\omega|\leqslant 2\\ 1 & 2.5\leqslant|\omega|\leqslant\pi\end{cases}$$

可以用 I 型和 III 型分别计算，并进行比较讨论。

7.23 设计一个三阶数字微分器，使得它的幅特性在 $\omega=\pi/4,\pi/2$ 和 $3\pi/4$ 的三个频点上与 $D(\omega)$相吻合。能否做到严格吻合？若做不到，用最小二乘求解。

7.24 用矩形窗函数设计一个六阶希尔波特变换器，再用哈明窗做同样设计，并比较之。

7.25 用 fir2 函数生成长度为 31 的一个窄带序列，其预期幅频特性为

$$D(\omega)=\begin{cases}0 & 0<|\omega|\leqslant\pi/4\\ 1 & \pi/4<|\omega|\leqslant\pi/3\\ 0 & \pi/3<|\omega|\leqslant\pi\end{cases}$$

分别采用矩形窗和汉宁窗实现。画出序列的实际幅频特性，与预期特性做比较并讨论。

第 8 章　IIR 滤波器设计

IIR 数字滤波器的设计和模拟滤波器的设计有着很紧密的关系。通常要先设计出适当的模拟滤波器，再通过一定的频带变换把它转换成为所需的数字 IIR 滤波器。此外，任何数字信号处理系统中也还不可避免地用到模拟滤波器，比如 A/D 变换器前的抗混叠滤波器和 D/A 变换器后的平滑滤波器，因此模拟滤波器设计也是数字信号处理中应当掌握的技术。在本章中，将首先介绍模拟滤波器的设计方法，然后讨论如何把模拟滤波器转换成为所需的数字 IIR 滤波器。

8.1　关于模拟滤波器的预备知识

8.1.1　模拟滤波器的指标

在本节中讨论两个问题，首先介绍平方幅度响应指标，它是模拟滤波器的基本指标，也是 IIR 滤波器的指标。通常把通带最大幅特性归一化为一，因而滤波器指标以相对的形式给出。设 $H_a(j\Omega)$ 是一个模拟滤波器的频率响应，则基于平方幅度响应 $J(\Omega)=|H_a(j\Omega)|^2$ 的低通滤波器技术指标为：

$$\begin{aligned}&\frac{1}{\sqrt{1+\varepsilon^2}}\leqslant|H_a(j\Omega)|^2\leqslant 1, &&|\Omega|<\Omega_p\\&0\leqslant|H_a(j\Omega)|^2\leqslant\frac{1}{A^2}, &&\Omega_s\leqslant|\Omega|\end{aligned}\tag{8.1.1}$$

其中 ε 为通带波动系数，Ω_p 和 Ω_s 是单位为[rad / s]的通带和阻带边缘频率。A 为阻带衰减参数。这些技术指标如图 8.1.1 所示。

从图中可知 $J(\Omega)$ 必须满足

$$|H_a(j\Omega)|^2=\begin{cases}\dfrac{1}{1+\varepsilon^2} & 当\Omega=\Omega_p\\ 1/A^2 & 当\Omega=\Omega_s\end{cases}\tag{8.1.2}$$

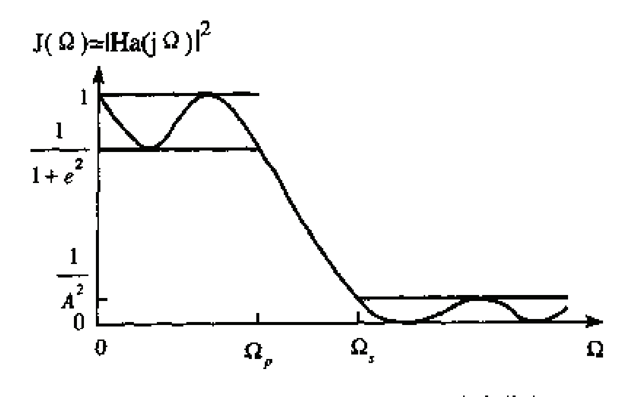


图 8.1.1　模拟低通滤波器技术指标

参数 ε 和 A 化成 dB 值后可与第 7 章中的数字滤波器指标 R_p 和 A_s 建立关系。其关系式为：

$$R_\mathrm{p}=-10\log_{10}\frac{1}{1+\varepsilon^2} \Rightarrow \varepsilon=\sqrt{10^{Rp/10}-1} \tag{8.1.3a}$$

$$A_\mathrm{s}=-10\log_{10}\frac{1}{A^2} \Rightarrow A=10^{As/20} \tag{8.1.3b}$$

绝对波动值 δ_1、δ_2 与 ε 和 A 之间又如下关系：

$$\frac{1-\delta_1}{1+\delta_1}=\sqrt{\frac{1}{1+\varepsilon^2}} \Rightarrow \varepsilon=\frac{2\sqrt{\delta_1}}{1-\delta_1} \tag{8.1.4a}$$

$$\frac{\delta_2}{1+\delta_1}=\frac{1}{A} \Rightarrow A=\frac{1+\delta_1}{\delta_2} \tag{8.1.4b}$$

从（8.1.1）式可知，模拟滤波器技术指标以平方幅度响应的形式给出，所以它不包含相位信息。为了计算 s 域系统函数 $H_\mathrm{a}(s)$，要把平方幅度响应表示为 s 的函数。从数学上看，要由复平面上一根虚轴（频率轴）上的函数，推算出全复平面上的函数，那是没有惟一解的。为此要作两点限制：一是规定函数的结构形式为 s 的实有理分式，所以它对实轴对称；二是此函数要对虚轴对称。这样，考虑到

$$H_\mathrm{a}(\mathrm{j}\Omega)=H_\mathrm{a}(s)\big|_{s=\mathrm{j}\Omega}$$

因此，有

$$|H_\mathrm{a}(\mathrm{j}\Omega)|^2=H_\mathrm{a}(\mathrm{j}\Omega)H_\mathrm{a}^*(\mathrm{j}\Omega)=H_\mathrm{a}(\mathrm{j}\Omega)H_\mathrm{a}(\mathrm{j}\Omega)=H_\mathrm{a}(s)H_\mathrm{a}(-s)\big|_{s=\mathrm{j}\Omega}$$

或者

$$H_\mathrm{a}(s)H_\mathrm{a}(-s)=|H_\mathrm{a}(\mathrm{j}\Omega)|^2_{\Omega=s/\mathrm{j}} \tag{8.1.5}$$

于是模拟滤波器设计问题可以表述如下。设它的传递函数为

$$H_\mathrm{a}(s)=\frac{B(s)}{A(s)}=\frac{b_0s^N+b_1s^{N-1}+\cdots+b_N}{a_0s^N+a_1s^{N-1}+\cdots+a_N} \tag{8.1.6}$$

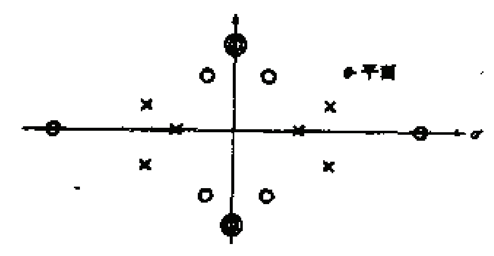
要寻求一个有理且稳定的 $H_\mathrm{a}(s)$ 使其平方幅频特性

$$\mathrm{j}(\Omega)=|H_\mathrm{a}(\mathrm{j}\Omega)|^2=H_\mathrm{a}(\mathrm{j}\Omega)H_\mathrm{a}^*(\mathrm{j}\omega)=H_\mathrm{a}(s)H_\mathrm{a}(-s)\big|_{s=\mathrm{j}\Omega}=\frac{B(s)B(-s)}{A(s)A(-s)}\bigg|_{s=\mathrm{j}\Omega} \tag{8.1.7}$$

满足条件（8.1.2）式。由（8.1.7）式可见，$J(\Omega)$ 的分解非常容易。只要它是 Ω^2 的有理分式：

$$J(\Omega)=J(\Omega^2)=\frac{D(\Omega^2)}{C(\Omega^2)}=\frac{d_0\Omega^{2N}+d_1\Omega^{2N-2}+\cdots+d_N}{c_0\Omega^{2N}+c_1\Omega^{2N-2}+\cdots+c_N} \tag{8.1.8}$$

此外还要求一些附加条件。因为 $s=\mathrm{j}\Omega$，所以 $s^2=-\Omega^2$。将它代入上式，于是 $D(-s^2)$ 和 $C(-s^2)$ 成为 s 的偶多项式。如果 λ 是 $D(-s^2)$ 或 $C(-s^2)$ 的根，那么 $-\lambda$ 也是一个根，即根必然对称于虚轴。另外由于 $D(-s^2)$ 和 $C(-s^2)$ 都是实系数的，所以它们的根必须共轭出现，因而又必然对称于实轴；这样 $D(-s^2)$ 和 $C(-s^2)$ 的根就四个为一组地在四个象限中镜像对称地出现，如图 8.1.2 所示。$D(-s^2)$ 的根可以出现在虚轴上，但由于上述理由，它们必须是偶数重根。而为了系统的稳定性，$C(-s^2)$ 就不得有在虚轴上的根。



注：让左半平面的零极点构成 Ha(s)，就得到最小相位系统

图 8.1.2 典型的 Ha(s)Ha(-s)零极点模式

在这样的条件下，$J(\Omega)$ 就可以按（8.1.7）式分解因式。分解为 $A(s)$和 $B(s)$的方法有很多种，由于我们要求最后的 $H(s)$是稳定的，$A(s)$必须包含 $J(\Omega)$ 在左半平面所有的极点；如果要求 $Ha(s)$是最小相位系统，则 $B(s)$必须包含 $J(\Omega)$ 在左半平面所有的零点。所以，设计一个模拟滤波器来达到给定的幅特性是比较直观而方便的。只要先找到一个满足幅特性要求(8.1.3)式的 Ω^2 的有理真分式，然后取出它的左半平面零点 z_j 和极点 p_k，用这些零极点可以组成一个稳定的模拟滤波器。

$$H_a(s)=\prod_{\mathrm{real}(zj,pk)<0}\frac{(s-z_j)}{(s-p_k)} \tag{8.1.9}$$

在这样的意义下，模拟滤波器的传递函数 $H_a(s)$可由平方幅度函数指标直接设计出来。

8.1.2 以模拟滤波器为基础的设计方法

模拟滤波器具有无限长的脉冲响应，IIR 数字滤波器也具有无限长脉冲响应，因此它与模拟滤波器相似。IIR 滤波器设计的基本技术就是利用复变函数映射把模拟滤波器转换成数字滤波器。这种技术的优点是：模拟滤波器设计（Analog Filter Design，AFD）在许多著作中都有现成的论述和列表，设计方法已很成熟。而模拟到数字滤波器的复值映射也是成熟的，许多文献中都论述过此映射。不过，只有低通滤波器的 AFD 表是现成的。为了设计其他的选频滤波器（高通，带通，带阻等），需要对低通滤波器进行频带变换，所以频带变换也是 IIR 滤波器设计的基本技术之一，在设计过程中要遇到两种不同的变换：(1)频带变换；(2)模拟/数字变换。根据这两种变换的先后次序，引出两种设计方法。

方法 1 如图 8.1.3(a)所示，方法 2 如图 8.1.3(b)所示。

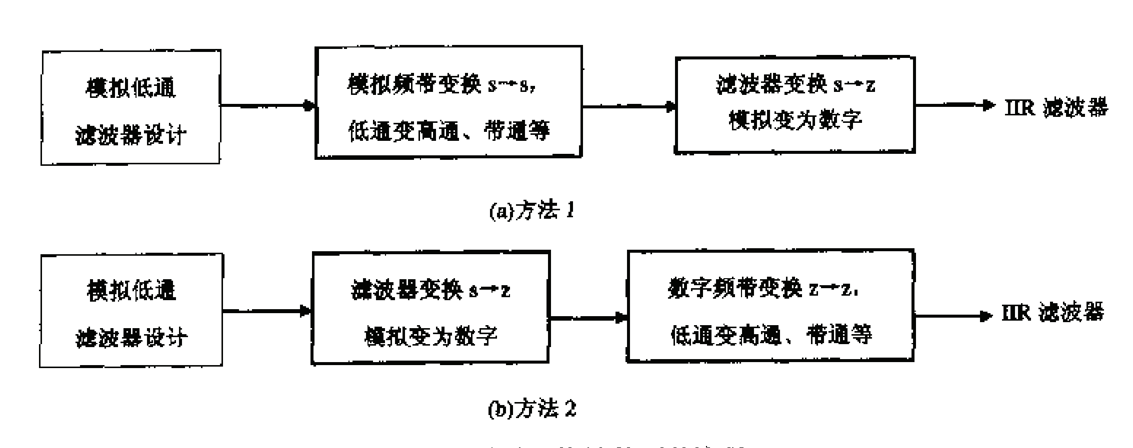


图 8.1.3　IIR 滤波器设计的两种流程

在 MATLAB 的信号处理工具箱中，采用第一种方法。使用这种方法，可以直接调用它所

提供的函数，比较方便。但如果不求甚解地直接运用这些工具箱函数，就不能透彻地掌握设计方法。因此，希望读者注意弄清道理。在这种 IIR 滤波器设计技术中，遵循以下步骤：

（1）设计模拟低通滤波器；

（2）进行频带变换，从模拟低通滤波器变换，得到高通、带通等模拟滤波器；

（3）进行模拟/数字滤波器变换，得到对应类型的数字滤波器。

这种滤波器设计方法存在的主要问题是不能控制 IIR 滤波器的相位特性，它只考虑了幅度进行设计。更成熟的技术可以同时逼近幅度和相位，它们需要更先进的优化工具箱，本书里不涉及这些内容。

下节讨论如何由平方幅度响应特性引出三种广泛应用的模拟低通滤波器，即巴特沃斯、切比雪夫和椭圆滤波器。以后的各节将研究怎样把低通模拟滤波器原型转换成其他选频特性的模拟滤波器，以及怎样把模拟滤波器变换为数字滤波器的问题。

8.2　模拟滤波器原型的设计

IIR 滤波器设计技术从已知的低通模拟滤波器出发，我们把这些模拟滤波器叫做滤波器原型。实际中有三种广泛应用的滤波器原型，即巴特沃斯低通、切比雪夫低通以及椭圆低通滤波器。在本节中，简要叙述这些低通滤波器原型的特征。尽管 MATLAB 信号处理工具箱提供了设计这些滤波器的函数，但是还是有必要知道这些滤波器的数学描述和主要特征，以便在 MATLAB 函数中选择合适的参数，以得到正确的结果。

8.2.1 巴特沃斯低通滤波器

这种滤波器的特征是其通带和阻带都有平坦的幅度响应。N 阶低通滤波器的平方幅度响应的表达式为：

$$J(\Omega)=\left|H_{\mathrm{a}}(\mathrm{j}\Omega)\right|^{2}=\frac{1}{1+\left(\dfrac{\Omega}{\Omega_{\mathrm{c}}}\right)^{2N}} \tag{8.2.1}$$

其中，N 为滤波器的阶数，Ω_c 是边缘频率（单位为 rad /s）。取 N=1,2,10，可得到平方幅频响应，如图 8.2.1 所示。

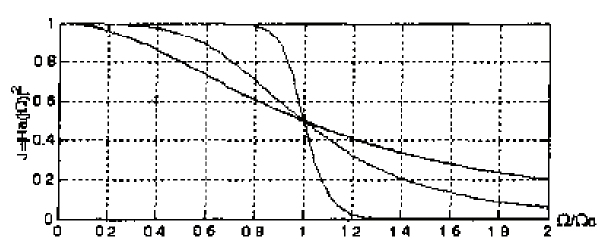


图 8.2.1 巴特沃思低通滤波器的平方幅频特性

从图 8.2.1 中可以观察到下列性质：

（1）当 $\Omega=0$ 时，$|H_a(\mathrm{j}\,\Omega)|^2=1$；

（2）$\Omega=\Omega_c$ 时，$|H_a(\mathrm{j}\,\Omega)|^2=1/2$，这说明在 Ω_c 处有 3 dB 的衰减；

（3）当 $\Omega\rightarrow\infty$ 时，$|H_a(\mathrm{j}\,\Omega)|^2=0$；

（4）$|H_a(\mathrm{j}\,\Omega)|^2$ 是 Ω 的单调减函数；

（5）$N\to\infty$时，$|H_a(j\Omega)|^2$为趋于理想的低通滤波器；

（6）在 $\Omega=0$ 处，$|H_a(j\Omega)|^2$ 对 Ω的各阶导数都存在，且等于 0，因此在这点上$|H_a(j\Omega)|^2$取极值。

为了确定系统函数 $H_a(s)$，把（8.2.1）式代入（8.1.5）式,进行零极点分解，得到：

$$H_a(s)H_a(-s)=\frac{1}{1+\left(\frac{s}{j\Omega_c}\right)^{2N}}=\frac{(j\Omega_c)^{2N}}{s^{2N}+(j\Omega_c)^{2N}} \tag{8.2.2}$$

上式分母多项式的特征方程为$s^{2N}+(j\Omega_c)^{2N}=0$，它的根就是系统的极点。注意到$\sqrt[2N]{-1}$的解是幅度为 1，相角为把 π 等分为 2$N$ 份的 N 个单位向量，可得这 2N 个根为

$$p_k=(j\Omega_c)\cdot(-1)^{1/2N}=\Omega_c e^{j\pi(2k+N+1)/2N}\qquad k=0,1,\cdots,2N-1 \tag{8.2.3}$$

对（8.2.3）式可做如下说明：

（1）$H_a(s)H_a(-s)$有 2N 个极点，等间隔分布在半径为 Ω_c 的圆上，其角度间隔为 π/N rad；

（2）N 为奇数时，极点为 $p_k=\Omega_c e^{j\pi k/N}$，$k=0,1,\cdots,2N-1$；

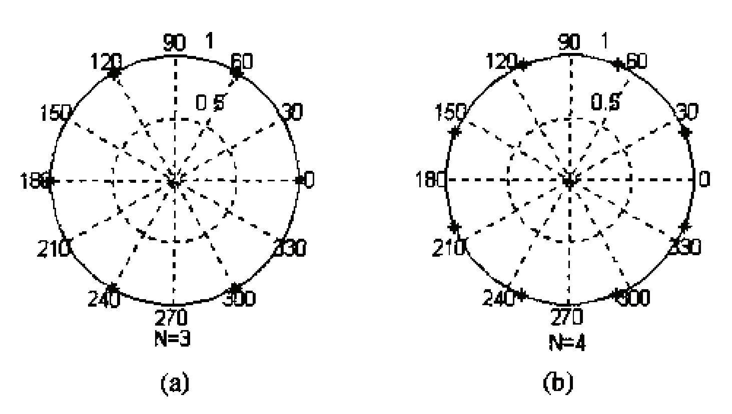
（3）N 为偶数时，极点为 $p_k=\Omega_c\, e^{j(k+1/2)\pi/N}$，$k=0,1,\cdots,2N-1$；

（4）极点关于 jΩ对称；

（5）极点绝不会落在虚轴上；

（6）此滤波器没有零点。

图 8.2.2 的(a)、(b)两个子图分别给出了三阶和四阶巴特沃斯滤波器的平方幅特性的极点分布图。



(a)　　(b)

图 8.2.2　巴特沃思低通滤波器平方幅特性的极点分布

为了求出此滤波器的系统函数，选择$H_a(s)H_a(-s)$左半平面的全部 N 个极点，得到一个稳定因果的滤波器$H_a(s)$。对于 N=3，表达式为：

$$H_a(s)=\frac{1}{\prod\limits_{real(pk)<0}(s-p_k)}=\frac{\Omega_c^3}{(s-\Omega_c e^{j2\pi/3})(s-\Omega_c e^{j\pi})(s-\Omega_c e^{j2\pi/3})} \tag{8.2.4}$$

例 8.2.1：给定$|H_a(j\Omega)|^2=\dfrac{1}{1+64\Omega^6}$，试确定模拟滤波器的系统函数$H_a(s)$。

解：根据给定的平方幅度响应得：

$$|H_a(j\Omega)|^2 = \frac{1}{1+64\Omega^6} = \frac{1}{1+\left(\dfrac{\Omega}{0.5}\right)^{2\times3}} \tag{8.2.5}$$

与（8.2.2）式比较，得到 $N=3, \Omega_c=0.5$。$Ha(s)Ha(-s)$的极点如图 8.2.2(a)所示。因此

$$H_a(s) = \frac{0.5^3}{(s+0.25-j0.433)(s+0.5)(s+0.25+j0.433)} = \frac{0.125}{(s+0.5)(s^2+0.5s+0.25)} \tag{8.2.6}$$

MATLAB 提供了一个叫做 buttap(Butterworth Analog Prototype)的函数，来设计 N 阶归一化（即 $\Omega_c=1$）巴特沃斯模拟原型滤波器，其调用格式为[z0,p0,k0]=buttap(N)。只要输入巴特沃斯滤波器的阶数 N，它可返回零点和极点数组 $z0$ 和 $p0$ 以及增益 $k0$。当需要具有任意 Ωc 的未归一化的巴特沃斯滤波器时，就要用 Ω_c 乘以 $p0$，极点有 N 个，所以分子也要乘以 Ω_C^N 由于没有零点就用 Ω_c^N 乘以 $k0$，以得到非归一化的 p 和 k。例 8.2.1 的滤波器的极点就位于半径为 0.5 的圆上，即 Ω_c=0.5。计算本例的程序 hc821 应为：

```
N=input('N= ');                        % 输入滤波器参数 N 和 OmegaC
OmegaC=input('OmegaC= ');
[z0,p0,k0] = buttap(N)                 % 调用buttap 函数,得到系统的左半平面零极点
p = p0*OmegaC; z = z0*OmegaC;          % 将零极点乘以 OmegaC，得到非归一化零极点
k = k0*OmegaC^N;                       % 将 k0 乘以 OmegaC^N,得到非归一化 k
b = real(poly(z));b = k*b              % 由零点计算分子系数向量
a = real(poly(p))                      % 由极点计算分母系数向量
```

上述程序中带波纹线的四行语句是用来求直接形式结构系数向量 b 和 a 的。如果想变为二阶实系数环节的级联，可以利用第 6 章中介绍的把 z 变换传递函数的直接形式转换成级联形式的 MATLAB 函数 tf2sos。这个函数是针对以 z 的负幂排列的多项式开发的。虽然可以推广到 s 域，但连续系统传递函数是按 s 的正幂排列的，要使两者一致，关键是使分子分母系数向量 b 和 a 同长，两序列中各元素的幂次排列一致。在本例中，可以在上面的程序最后加上以下两条语句，则可得出级联二届环节的参数。

```
bP=[zeros(1,length(a)-length(b)),b];  % 将b左端补零，使与a 同长，称为 bp
[sos,G]=tf2sos(bP,a)                   % 求分解为级联二阶环节的参数
```

运行此程序，并输入 N=3 和 OmegaC=0.5，得到

```
b =    0.1250
a =    1.0000    1.0000    0.5000    0.1250
bp=    0.1250    0         0         0
sos =  0         1.0000    0         1.0000    0.5000    0
       0         0         1.0000    1.0000    0.5000    0.2500
G =    0.1250
```

这意味着传递函数的直接形式和二阶级联形式分别为：

$$H_a(s) = \frac{0.125}{s^3+s^2+0.5s+0.125} \quad 和 \quad H_a(s) = \frac{0.125}{(s+0.5)(s^2+0.5s+0.25)}$$

不言而喻，这两个式子是等价的。

设计过程中，通常需要由给定的滤波器指标参数 Ω_p, R_p, Ω_s 和 A_s，求得滤波器的阶数 N 和边缘频率 Ω_c，这就要求把巴特沃斯低通滤波器的公式实用化，变换为它的设计方程。此方程可以推导如下。

如果要求滤波器在通带和阻带边缘频率上的幅特性满足指标 R_p 和 A_s，即满足下列方程：

$$\Omega=\Omega_p \text{时}，\ -10\log_{10}|H_a(j\Omega_p)|^2=R_p$$

$$\Omega=\Omega_s \text{时}，\ -10\log_{10}|H_a(j\Omega_s)|^2=A_s$$

这样就有两个联立方程

$$-10\log_{10}\left(\frac{1}{1+\left(\Omega_p/\Omega_c\right)^{2N}}\right)=R_p \tag{8.2.7 a}$$

$$-10\log_{10}\left(\frac{1}{1+\left(\Omega_s/\Omega_c\right)^{2N}}\right)=A_s \tag{8.2.7 b}$$

将这两个方程对 N 和 Ω_c 求解，并取 N 为比计算结果大的最小整数，得：

$$N=\left\lceil\frac{\log_{10}\left[(10^{Rp/10}-1)/(10^{As/10}-1)\right]}{2\log_{10}(\Omega_p/\Omega_s)}\right\rceil \tag{8.2.7}$$

由于选择的 N 比所需要的略大，如果在 Ω_p 处满足指标，则 Ω_s 处将有裕度，反之亦然。若要在 Ω_p 处精确满足指标，要求

$$\Omega_{cp}=\frac{\Omega_p}{\sqrt[2N]{10^{Rp/10}-1}} \tag{8.2.8 a}$$

而为了在 Ω_s 处精确满足指标，则要求

$$\Omega_{cs}=\frac{\Omega_s}{\sqrt[2N]{10^{As/10}-1}} \tag{8.2.8 b}$$

实际设计时，Ω_c 可在 $\Omega_{cp}\leqslant\Omega_c\leqslant\Omega_{cs}$ 范围内选择。式(8.2.7)和(8.2.8)就是巴特沃斯低通滤波器的设计方程。写成 MATLAB 语句为

```
N=ceil( log10((10.^(0.1*abs(Rp))-1)./(10.^(0.1*abs(As)) -1))/(2*log10(OmegaP/OmegaS)) );
OmegaC= OmegaP / ( (10^(.1*abs(Rp)) - 1)^(1/(2*N)) );
```

以这两条语句为核心，在信号处理工具箱中已编成一个函数 buttord，它可根据给定的滤波器指标 Ω_p, R_p, Ω_s 和 A_s，求得滤波器的阶数 N 和边缘频率 Ω_c，其调用格式为：

[N, OmegaC] =buttord(OmegaP, OmegaS, Rp, As, 's') (8.2.9)

其中 OmegaP,OmegaS,OmegaC 均以 rad / s 为单位，R_p 和 A_s 以 dB 为单位，最后的输入变元"*s*"说明所设计的是模拟滤波器。如果少了这个变元，MATLAB 会以为是直接设计数字滤波器，结果就完全不对了。

例 8.2.2：设通带边缘频率 $f_p=5$ kHz，最大衰减 R_p 为 1 dB，阻带边缘频率 f_s=12 kHz，最大衰减 A_s 为 30 dB，要求设计巴特沃斯低通滤波器。

解：用手工运算的方法可按下列步骤：

（1）用式(8.2.7)和式(8.2.8)确定阶数 N 及边缘频率 Ω_c：

$$N=\left\lceil\frac{\log_{10}\left[(10^{1/10}-1)/(10^{30/10}-1)\right]}{2\log_{10}^{(5/12)}}\right\rceil=5$$

按(8.2.8 a) $\Omega_{cp}=\dfrac{2*\pi*5000}{\sqrt[10]{10^{1/10}-1}}=33147$，按(8.2.8 b) $\Omega_{cs}=\dfrac{2*\pi*12000}{\sqrt[10]{10^{30/10}-1}}=37792$。

实际的 Ω_c 可以选在两者之间的任何值；

（2）用(8.2.3)式，根据 N 值求出极点 R_k 分布，取出其左半平面全部极点；

（3）用(8.2.4)式求出滤波器的系数向量。

不需说明，这个计算是相当繁琐的。过去的书上列出了表格，来避免（2）和（3）两项计算。但通常也只能列到 $N<10$ 的范围，本书则可直接调用 buttap(N)函数，写出下列 hc822 程序：（若直接调用 MATLAB 设计公式 buttord，则语句更简单，也列于该程序中）

```
OmegaP=2*pi*5000;OmegaS=2*pi*12000;
Rp=1;As=30;
% 确定滤波器的阶次和边缘频率
N=ceil(log10((10.^(0.1*abs(Rp))-1)./(10.^(0.1*abs(As)) -1)) /
(2*log10(OmegaP/OmegaS)) )
OmegaC= OmegaP / ( (10^(.1*abs(Rp)) - 1)^(1/(2*N)) );     % 确定滤波器的Ωc
% [N, OmegaC] =buttord(OmegaP, OmegaS, Rp, As, 's')     % 上两条语句可用这条语句代替
[z0,p0,k0] = buttap(N)                    % 调用buttap 函数,得到系统的左半平面零极点
b0=k0*real(poly(z0)); a0=real(poly(p0))    % 求归一化滤波器系数b0,a0
加上hcg21 中带波纹线的语句，可求出非归一化的滤波器个数
```

运行此程序，得到

```
N = 5
OmegaC = 3.7792e+004
z0 =    [ ]
p0 =  -0.3090 + 0.9511i
  -0.3090 - 0.9511i
  -0.8090 + 0.5878i
  -0.8090 - 0.5878i
  -1.0000
k0 =    1
b = 7.7094e+022
a = 1.0e+022 *[ 0.0000   0.0000   0.0000   0.0000   0.0007   7.7094]
```

可以看出，由于 Ω_c 相当大，达到 37792[rad / s]。分子分母上出现的 $\Omega_c{}^N$ 达到 10^{22} 的数量级，以致向量 a 中其他较小的系数无法显示。因此最好仍用归一化的系数表示滤波器参数，而把变量 s 写成（$s0= s/\ \Omega_c$），这时自变量仍为 s，所以仍是非归一化形式。得出：

```
b0 = 1
a0 =  1.0000  3.2361  5.2361  5.2361   3.2361  1.0000
```

这样，巴特沃斯滤波器传递函数可写成如下易于准确表达的非归一化形式。

$$H_a(s)=\frac{1}{\left(\dfrac{s}{37792}\right)^5+3.2361\left(\dfrac{s}{37792}\right)^4+5.2361\left(\dfrac{s}{37792}\right)^3+5.2361\left(\dfrac{s}{37792}\right)^2+3.2361\left(\dfrac{s}{37792}\right)+1}$$

也可写成 $H_a(s)=\dfrac{1}{s0^5+3.2361s0^4+5.2361s0^3+5.2361s0^2+3.2361s0+1}$。

其中：$s0=\left(\dfrac{s}{37792}\right)$。

为了检验此滤波器在 Ω_p 和 Ω_s 处的幅特性，可以调用连续系统频率特性计算函数 freqs，它是 freqz 函数在连续频域的对等物。其典型调用格式为：

H =freqs(b,a,w)

其中：b 为滤波器分子系数向量；

a 为滤波器分母系数向量；

w 为要取做自变量的频率向量，不允许取标量，故至少要同时算两个频点。

如果 b 和 a 取归一化形式 $b0$ 和 $a0$，而频率也取归一化形式 w0=w/OmegaC，即用

H =freqs(b0,a0,w/OmegaC)

其结果与全用上面语句中用非归一化参数的结果相同。如果对该取的频率范围 w 心中无底，可以用 freqs 函数的另一种调用格式：

[H,w] =freqs(b,a)

此时系统会自动在适当的频率范围内选择 200 个频点进行计算，用 plot(w,abs(H))即可画出其频率响应。

在本例中，就取 OmegaP,OmegaS 两个频点组成计算向量 wx=[OmegaP,OmegaS]。因为本例用归一化系数 b0,a0，变量 w 也必须进行归一化，即把模拟频率除以 Ω_c，才能得到正确结果。于是键入以下语句：

```
wxo=[5000, 12000]*2*pi/37792;             % 设置频率向量 wxo=[OmegaP,OmegaS]/OmegaC
Hx=freqs(b0,a0,wxo)                       % 计算该两点上的幅特性
dbHx=-20*log10(abs(Hx)/max(abs(H)))       % 化为分贝值
```

得到
```
Hx =  -0.9269 - 0.0684i   0.0314 + 0.0035i
dbHx =   1.0000   32.1555
```

可见滤波器在 OmegaP 处的 Rp=1 dB，在 OmegaS 处的 As=32 dB，满足了设计要求。

键入无左端变量的语句 freqs(b,a)，MATLAB 会自动给出系统的波德图，因为输入变元取的是 a 和 b，结果是实际的非归一化的频率特性，如图 8.2.3 所示。关于 freqs 的其他用法，可用 help freqs 语句得知。

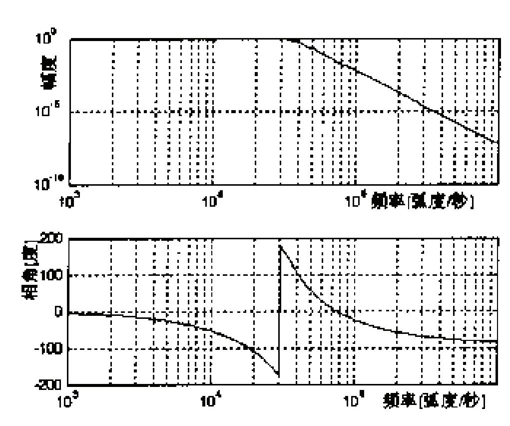


图 8.2.3　例 8.2.2 的滤波器幅频和相频特性

8.2.2 切比雪夫 I 型低通滤波器

巴特沃斯滤波器在通带和阻带上的响应都是单调的。而切比雪夫滤波器具有波动性。它有两种类型：切比雪夫 I 型滤波器在通带中具有等波动响应，而切比雪夫 II 型在阻带中具有等波动响应。回忆讨论过的等波动 FIR 滤波器，可知等波动特性比单调特性的滤波器有较低的阶次。因此，对于相同的指标，切比雪夫滤波器比巴特沃斯滤波器的阶数低。

切比雪夫 I 型滤波器的平方幅度响应为：

$$J(\Omega)=\left|H_{\mathrm{a}}(\mathrm{j}\Omega)\right|^{2}=\frac{1}{1+\varepsilon^{2}T_{N}^{2}(\Omega/\Omega_{c})} \tag{8.2.10}$$

其中 N 为滤波器的阶数，ε 为通带波动系数，它与 R_p 有关，$T_N(x)$是 N 阶切比雪夫多项式。

$$T_N(x)=\begin{cases}\cos(N\cos^{-1}(x)) & 0\leqslant x\leqslant 1\\ \cosh(\cosh^{-1}(x)) & 1<x<\infty\end{cases} \tag{8.2.11}$$

切比雪夫滤波器的等波动响应是由多项式 $T_N(x)$ 造成的。它的主要性质是：

（1）$0<x<1$ 时，$T_N(x)$ 出现在 -1 至 1 之间的振荡；

（2）$0<x<\infty$时，$T_N(x)$单调增至∞。

$|H_a(\mathrm{j}\Omega)|$ 在 $\Omega=0$ 处有两种可能的取值，一种是 N 为奇数时，一种是 N 为偶数时，如图 8.2.4 所示。注意 $x=\Omega/\Omega_c$为归一化的频率。

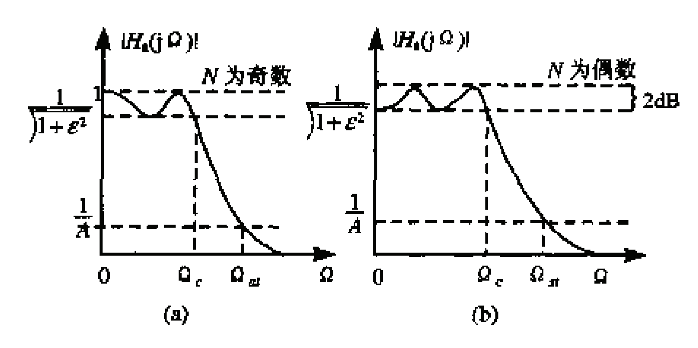


图 8.2.4　切比雪夫 I 型滤波器的幅度特性

从图 8.2.4 可观察到下列性质：

（1）$x=0$（或 $\Omega=0$）时，$|H_a(0)|=1$（N 为奇数）或$|H_a(0)|=1\big/\sqrt{1+\varepsilon^2}$（$N$ 为偶数）；

（2）$x=1$（或 $\Omega=\Omega_c$）时，$|H_a(1)|=1\big/\sqrt{1+\varepsilon^2}$（对所有的 N）；

（3）$0\leqslant x\leqslant 1$（或 $0\leqslant\Omega\leqslant\Omega_c$）时，$|H_a(\mathrm{j}x)|$在 1 和$1\big/\sqrt{1+\varepsilon^2}$ 之间振荡；

（4）$x>1$（或 $\Omega>\Omega_c$）时，$|H_a(\mathrm{j}x)|$单调减至 0；

（5）$x=\Omega_c$时，$|H_a(\mathrm{j}x)|=1/A$。

为了确定一个因果的和稳定的 $H_a(s)$，必须找出 $H_a(s)H_a(-s)$ 的极点，它们应该是 $1+\varepsilon^2T_N^2\left(\dfrac{s}{\mathrm{j}\Omega_c}\right)=0$ 的根。然后把左半平面的极点分配给 $H_a(s)$。

求解这个方程式是比较困难而冗长的。可以证明，如果 $p_k=\sigma_k+\mathrm{j}\Omega_k$，$(k=0,\ldots,N-1)$是上述多项式的左半平面根，则

$$\sigma_k=(a\Omega_c)\cos\left[\frac{\pi}{2}+\frac{(2k+1)\pi}{2N}\right]$$

$$k=0,1,\cdots,N-1 \tag{8.2.12}$$

$$\Omega_k=(a\Omega_c)\sin\left[\frac{\pi}{2}+\frac{(2k+1)\pi}{2N}\right]$$

其中：$a=(\sqrt[N]{x}-\sqrt[N]{1/x})/2,\ b=(\sqrt[N]{x}+\sqrt[N]{1/x})/2,\ x=\dfrac{1}{\varepsilon}+\sqrt{1+\dfrac{1}{\varepsilon^2}}$。

这些根落在长轴为 $b\,\Omega_c$，短轴为 $a\,\Omega_c$ 的椭圆上。如图 8.2.5 所示。

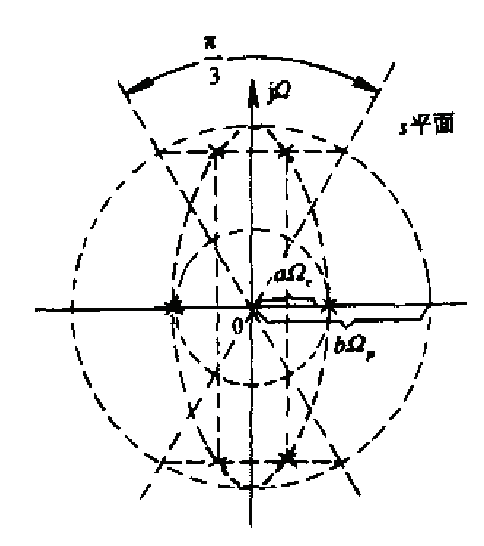


图 8.2.5 三阶切比雪夫滤波器的极点分布

切比雪夫 I 型滤波器也没有零点，因此滤波器的系统函数按式（8.1.8）为

$$H_a(s)=\prod_{\mathrm{real}(pk)<0}\frac{k}{(s-p_k)} \tag{8.2.13}$$

其中 k 是归一化因子，它使得

$$H_a(0)=\begin{cases}1\\ 1\Big/\sqrt{1+\varepsilon^2}\end{cases} \tag{8.2.14}$$

最后得到的归一化传递函数为

$$H_a(s)=\frac{1}{\varepsilon\bullet 2^{N-1}\prod_{k=1}^{N}(\frac{s}{\Omega_c}-\pi P_{k0})} \tag{8.2.15 a}$$

去归一化的传递函数为

$$H_a(s)=\frac{\Omega_c^N}{\varepsilon\bullet 2^{N-1}\prod_{k=1}^{N}(s-\Omega_c P_{k0})} \tag{8.2.15 b}$$

由于切比雪夫滤波器的计算公式比巴特沃斯滤波器还要繁琐，显然必须丢开手工运算而利用计算机协助。MATLAB 提供了一个函数[z0,p0,k0]=cheblap(N,Rp)，来设计一个阶数为 N，通带波动为 Rp 的归一化切比雪夫 I 型原型滤波器，它在数组 z0 中返回零点，数组 p0 中返回极点，并且返回增益 k0。若需要具有任意 Ω_c 的未归一化的切比雪夫 I 型滤波器。可以由归一化滤波器的极点数组 p0 乘以 Ω_c 得到。因为这种滤波器也没有零点。新增益 k 由 Ω_c^N 确定。所以要调用这个函数，必须先知道滤波器的阶数 N，通带波动 R_p 和边缘频率 Ω_c。用程序表述非归一化与非归一化滤波器零极点和系数向量之间的关系如下：

```
[z0,p0,k0]=cheblap(N,Rp)        % 求归一化切比雪夫 I 型模拟滤波器零极点
```

后面接着 hc821 或 hc822 中带下画线的语句，就可得到非归一化的滤波器参数 b 和 a。用它们就可以调用计算频率特性的函数[H,w]=freqs(b,a)以便计算系统的频率特性。

和巴特沃斯滤波器的情况相仿，非归一化的滤波器的各参数通常数量级相差很大，从显示和表达的角度看，更好的方法是用归一化形式表示传递函数，而把传递函数中的 s 用 $s\ /\ \Omega_c$ 表

示，这样在程序中可以省去那些带波纹的语句，同样也可以得到非归一化形式的滤波器。

在实际设计时，还需要根据前面的公式(8.2.11)~(8.2.15)，导出切比雪夫 I 型原型滤波器的设计公式，以便根据给定的滤波器技术指标 Ω_p,R_p, Ω_s 和 A_s，一步一步地求得滤波器的阶数 N，波动系数 ε 和边缘频率 Ω_c。其公式如下：

$$\varepsilon = \sqrt{10^{0.1R_p} - 1}, \qquad A = 10^{0.05A_s}, \tag{8.2.16}$$

$$\Omega_c = \Omega_p, \qquad \lambda_s = \Omega_s / \Omega_p, \tag{8.2.17}$$

$$N = \left\lceil \frac{\log_{10}\left(g + \sqrt{g^2 - 1}\right)}{\log_{10}\left(\lambda_s + \sqrt{\lambda_s^2 - 1}\right)} \right\rceil \tag{8.2.18}$$

其中：

$$g = \sqrt{(A^2 - 1)} / \varepsilon \tag{8.2.19}$$

信号处理工具箱中把这些设计公式也归纳成一个函数 cheb1ord，其调用格式为：

[N, OmegaC] = cheb1ord(OmegaP, OmegaS, Rp, As, 's')　　(8.2.20)

它与式(8.2.9)的 buttord 调用格式相似。

例 8.2.3：按例 8.2.2 的指标，f_p=5 kHz，R_p=1dB，f_s=12 kHz，A_s=30 dB，要求设计切比雪夫 I 型低通滤波器。

解：首先计算需要的参数,按公式（8.2.16）～（8.2.19）：

$$\varepsilon = \sqrt{10^{0.1R_p} - 1} = \sqrt{10^{0.1} - 1} = 0.5088$$

$$A = 10^{0.05\times 30} = 10^{1.5} = 31.6228$$

$$\Omega_c = \Omega_p = 2\pi * 5000 = 31416$$

$$\lambda_s = 12000/5000 = 2.4$$

$$g = \sqrt{(A^2 - 1)} / \varepsilon = \sqrt{(31.6228^2 - 1)} / 0.5088 = 62.1206$$

$$N = \left\lceil \frac{\log_{10}\left(g + \sqrt{g^2 - 1}\right)}{\log_{10}\left(\lambda_s + \sqrt{\lambda_s^2 - 1}\right)} \right\rceil = \left\lceil \frac{\log_{10}\left(62.1206 + \sqrt{62.1206^2 - 1}\right)}{\log_{10}(2.4 + \sqrt{2.4^2 - 1})} \right\rceil = \left\lceil \frac{2.0942}{0.6610} \right\rceil = \lceil 3.1682 \rceil = 4$$

现在可以由(8.2.12)式求出 $H_a(s)$：

$$x = \frac{1}{\varepsilon} + \sqrt{1 + \frac{1}{\varepsilon^2}} = \frac{1}{0.5088} + \frac{\sqrt{1 + 0.5088^2}}{0.5088} = 4.1706$$

$$a = (\sqrt[N]{x} - \sqrt[N]{1/x})/2 = (\sqrt[4]{4.1706} - \sqrt[4]{1/4.1706})/2 = 0.3646$$

$$b = (\sqrt[N]{x} + \sqrt[N]{1/x})/2 = (\sqrt[4]{4.1706} + \sqrt[4]{1/4.1706})/2 = 1.0644$$

$H_a(s)$的 N=4 个极点为：

$$p_{0,3} = \left[a\cos\left(\frac{\pi}{2} + \frac{\pi}{8}\right) \pm \mathrm{j}b\sin\left(\frac{\pi}{2} + \frac{\pi}{8}\right)\right]\Omega_c = (-0.1395 \pm 0.9834\mathrm{j}) \times 5000 \times 2\pi$$

$$p_{1,2} = \left[a\cos\left(\frac{\pi}{2} + \frac{3\pi}{8}\right) \pm \mathrm{j}b\cos\left(\frac{\pi}{2} + \frac{3\pi}{8}\right)\right]\Omega_c = (-0.3369 \pm 0.4073\mathrm{j}) \times 5000 \times 2\pi$$

因此 $H_a(s)=\prod_{real(p_k)<0}\frac{K}{(s-p_k)}=\frac{b_0}{a_4\left(\frac{s}{\Omega_c}\right)^4+a_3\left(\frac{s}{\Omega_c}\right)^3+a_2\left(\frac{s}{\Omega_c}\right)^2+a_1\left(\frac{s}{\Omega_c}\right)+1}$

将这四个复数极点 p_k 代入并展开求 a_i 是十分麻烦的工作。当没有计算机时，这些系数可以用查表的方法来求。但它的计算也真够繁杂的了，在到了 21 世纪的今天，计算机已如此普及，还用手工计算是不可取的，用 MATLAB 程序 hc823 计算如下：

```
OmegaP=2*pi*5000;OmegaS=2*pi*12000;                    % 输入设计指标
Rp=1; As=30;
[N, OmegaC] = cheb1ord(OmegaP, OmegaS, Rp, As, 's') % 调用切比雪夫 I 设计函数
[z0,p0,k0] = cheb1ap(N,Rp)                             % 调用切比雪夫 I 原型函数，得到左半平面零极点
b0 =k0* real(poly(z0));                                % 由零点计算分子系数向量
a0 = real(poly(p0))                                    % 由极点计算分母系数向量
```

程序运行的结果为：

```
N =    4
OmegaC =  3.1416e+004
z0 =     [ ]
p0 =  -0.1395 + 0.9834i
  -0.3369 + 0.4073i
  -0.3369 - 0.4073i
  -0.1395 - 0.9834i
k0 =   0.2457
b0 =   0.2457
a0 =   1.0000    0.9528    1.4539    0.7426    0.2756
```

因此此切比雪夫 I 型模拟滤波器的传递函数为

$$H_a(s)=\frac{0.2457}{\left(\frac{s}{31416}\right)^4+0.9528\left(\frac{s}{31416}\right)^3+1.4539\left(\frac{s}{31416}\right)^2+0.7426\left(\frac{s}{31416}\right)+1}$$

为了求出（非归一化的）原滤波器的频率特性和画出其零极点分布图，可再加上下列语句

```
[Ha,w0]=freqs(b0,a0);                              % 求频率响应
subplot(1,2,1),
semilogx(w0*OmegaC,20*log10(abs(Ha)/max(abs(Ha)))),grid on  % 画出分贝频率特性
subplot(1,2,2), plot(p0*OmegaC,'x')              % 画出极点分布图
axis square, axis equal, grid on                 % 使x,y 等比例，以便看出极点的椭圆分布
```

程序画出的图形见图 8.2.6。可以估计出它的通带波动约为 1dB，在 OmegaS=12000*2π ≈75 400 处的阻带衰减约为 35dB。

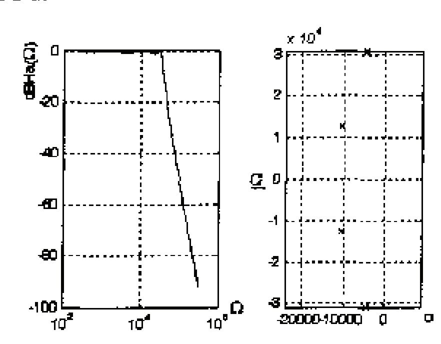


图 8.2.6 切比雪夫 I 型模拟低通滤波器原型的对数幅频特性和极点分布

要精确计算，可用语句

```
Hx = freqs(b0,a0,[OmegaP,OmegaS]/OmegaC);
dbHx=-20*log10(abs(Hx)/max(abs(Ha)))
```

得到 dbHx=　0.9998　40.9938，可见它能满足设计要求，并有相当的裕度。

如果要求出此滤波器的级联形式的传递函数，则如前面讨论巴特沃斯滤波器的情况一样，可以用 tf2sos 函数，只要注意分子分母系数向量必须取得同样长。

```
Nm=length(a0)-length(b0);                 % 求极点个数与零点个数之差
[sos,G]=tf2sos([zeros(1,Nm),b0],a0)       % 求级联形式的归一化传递函数
```

结果为

```
sos0 =   0       0    1.0000    1.0000    0.6737    0.2794
         0       0    1.0000    1.0000    0.2791    0.9865
G0 =     1
```

因而传递函数也可写成

$$H_a(s)=\frac{1}{\left(\left(\frac{s}{31416}\right)^2+0.6737\left(\frac{s}{31416}\right)+0.2794\right)\left(\left(\frac{s}{31416}\right)^2+0.2791\left(\frac{s}{31416}\right)+0.9865\right)}$$

8.2.3 切比雪夫 II 型低通滤波器

切比雪夫II型的平方幅度响应为：

$$J(\Omega)=\left|H_a(\mathrm{j}\Omega)\right|^2=\frac{1}{1+\left[\varepsilon^2T_N^2(\Omega_c/\Omega)\right]^{-1}} \tag{8.2.21}$$

切比雪夫II型滤波器通带幅特性是单调的，而阻带是等波动的，意味着这种滤波器在 s 平面上既有极点，又有零点。因此通带中的群延迟特征比切比雪夫 I 型滤波器好，也即相位响应更线性。如果把切比雪夫 I 型滤波器的平方幅度响应（8.2.10）式中的 $\varepsilon^2T_N^2(\Omega/\Omega_c)$ 代换为它的倒数，并且把变量 $x=\Omega/\Omega_c$ 也换为它自己的倒数，可把它转换成II型滤波器的平方幅度响应(8.2.21)。开方后得到的幅频特性画成曲线如图 8.2.7。

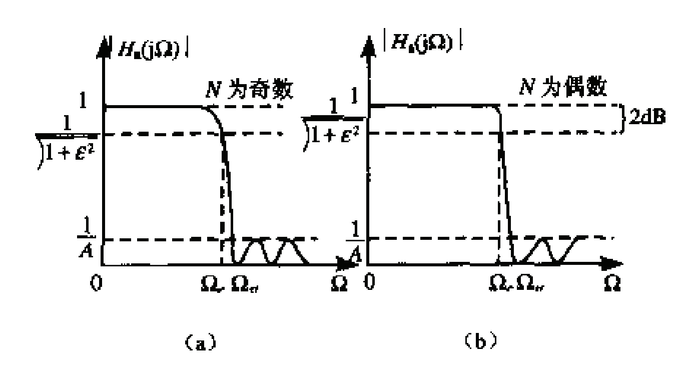


图 8.2.7　切贝雪夫 II 型滤波器的幅频特性

因为切比雪夫 II 型滤波器同时有零点和极点，它的公式推导比 I 型更加复杂，可直接利用计算机来进行设计。MATLAB 提供了一个函数[z0,p0,k0]=cheb2ap(N,As)，用来设计阶数为

N、阻带衰减为 As 的归一化切比雪夫II型原型滤波器，它在数组 $z0$ 中返回零点，数组 $p0$ 中返回极点，并且返回增益 $k0$。如果需要具有任意 Ω_c 的未归一化的切比雪夫II型滤波器，可把归一化滤波器的数组 $p0$ 乘以 Ω_c，由于它具有零点，数组 $z0$ 也必须乘以 Ω_c。新增益 k 由原增益 $k0$ 乘以 Ω_c^{Nm}，其中 Nm 为极点数减去零点数，用 MATLAB 语句表示为

Nm=length(p0) – length(z0)=length(a0) – length(b0)

因为波动是在阻带中，所以取 $\Omega_c=\Omega_s$。除此以外，切比雪夫II型原型的设计方程与切比雪夫 I 型类似。MATLAB 信号处理工具箱中也提供了一个 cheb2ord 函数，用它可以由给定的滤波器技术指标 Ω_p, R_p, Ω_s 和 A_s，求得滤波器的阶数 N 和边缘频率 Ω_c，其调用格式为

[N, OmegaC] = cheb2ord(OmegaP, OmegaS, Rp, As, 's')

正如前面巴特沃斯滤波器和切比雪夫 I 型滤波器一样，把这个函数中得出的 N 作为 cheb2ap 函数的输入变元，把两个函数衔接起来使用。同时 Ω_c 也已得到，就可以完成切比雪夫II型原型滤波器的设计问题。

例 8.2.4：按例 8.2.2 的指标，f_p=5 k Hz，R_p=1 dB，f_s=12 kHz，A_s=30 dB，要求设计切比雪夫 II 型低通滤波器来实现。

解：直接用 MATLAB 程序来解题，程序名为 hc824。该程序几乎是由 hc823 拷贝而得，只改变了几个变元：

```
OmegaP=2*pi*5000; OmegaS=2*pi*12000;        % 输入设计指标
Rp=1; As=30;
[N, OmegaC] = cheb2ord(OmegaP, OmegaS, Rp, As, 's') % 调用切比雪夫 II 设计函数
[z0,p0,k0] = cheb2ap(N, As)                 % 调用切比雪夫 II 原型函数,得到左半平面零极点
```

在得到 $p0,z0,k0$ 后，计算系统的传递函数归一化系数向量 $b0,a0$,绘制频率特性和零极点分布的语句和 hc823 中的完全相同，此处就予以省略。

程序运行后，得到如图 8.2.8 的曲线及数字结果。

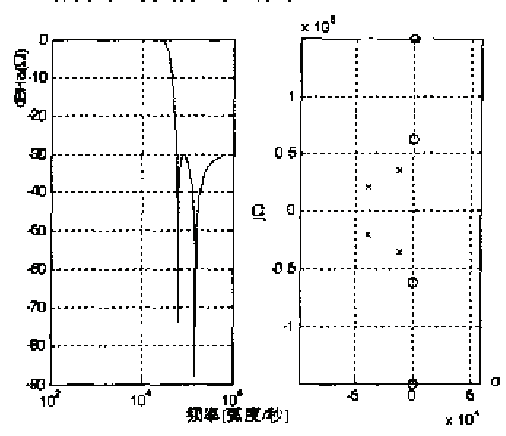


图 8.2.8 例 8.2.4 中切比雪夫 II 型模拟低通滤波器原型的对数幅频特性和零极点分布

```
N = 4
OmegaC =  5.7146e+004
z0 = 0 + 1.0824i
     0 - 1.0824i
     0 + 2.6131i
     0 - 2.6131i
p0 = -0.1988 - 0.6180i
     -0.6836 - 0.3646i
```

```
   -0.6836 + 0.3646i
   -0.1988 + 0.6180i
k0 = 0.0316
b0 = 1.0000     0     8.0000     0     8.0000
a0 = 1.0000   1.7648   1.5653    0.8149   0.2530
sos = 1.0000     0     6.8284    1.0000   0.3976   0.4214
    1.0000     0     1.1716    1.0000   1.3673   0.6003
G = 1
```

所以，所设计的滤波器的传递函数如下所示。

直接型：$H_a(s)=\dfrac{s0^4+s0^2+1}{s0^4+1.7648s0^3+1.5653s0^2+0.8149s0+0.2530}$

级联型：$H_a(s)=\dfrac{\left(s0^2+0.6848\right)\left(s0^2+1.1716\right)}{\left(s0^2+0.3976s0+0.4214\right)\left(s0^2+0.1.3673s0+0.6003\right)}$

其中 $s0=\dfrac{s}{57146}$ 为归一化拉普拉斯算子。

从图上可以看出切比雪夫Ⅱ型滤波器在阻带出现等幅波动的特性，而其峰值为 30 dB，达到了设计要求。同时也可以看到，本例中的切比雪夫Ⅱ型滤波器有四个零点，和极点的数目相等，它们都在虚轴上。

与前面两个例子比较，还可以看出，要达到同样的指标，用切比雪夫（包括Ⅰ型和Ⅱ型）方法设计出的滤波器为四阶，比用巴特沃斯方法设计出的滤波器低了一阶，因此它的效率较高。这是因为它充分利用了通带（或阻带）内允许波动的指标，多次接近 R_p 或 A_s，而巴特沃斯滤波器的幅特性是单调的，接近 R_p 或 A_s 的都只有一个频点。可以设想，如果能同时利用通带和阻带内的允许波动，有可能得到阶次更低的滤波器，椭圆滤波器就实现了这一设想。

8.2.4　椭圆低通滤波器

椭圆滤波器的通带和阻带均展现出等波动响应，它们的幅度响应特性与 FIR 等波动滤波器类似。因此对于给定的指标，它们可使阶数 N 最小。换言之，同样的阶数 N，它的过渡带最陡，从这个意义上讲，椭圆滤波器是最优滤波器。这种滤波器数学形式复杂，利用简单的工具不可能设计出它们，设计时经常需要现成的程序和表格。

椭圆滤波器的平方幅度响应为：

$$J(\Omega)=\left|H_a(\mathrm{j}\Omega)\right|^2=\frac{1}{1+\varepsilon^2U_N^2\left(\Omega/\Omega_c\right)} \tag{8.2.22}$$

其中 N 是阶数，ε 是通带波动（它与 R_p 有关），$U_N(\bullet)$ 是 N 阶雅可比椭圆函数，对此函数的分析，即使是很肤浅的，也超出了本书的范围。（8.2.22）式所描述的响应与切比雪夫滤波器有相似之处。N 为奇数和偶数时的典型响应如下图所示。

它的设计方法和前面所述的几种滤波器相仿，只是用到更复杂的函数。尽管很难对（8.2.22）式进行分析，但是计算阶数的公式却是很简洁的，它的计算公式为

$$N=\frac{K(k)K\left(\sqrt{1-k_1^2}\right)}{K(k_1)K\left(\sqrt{1-k^2}\right)} \tag{8.2.23}$$

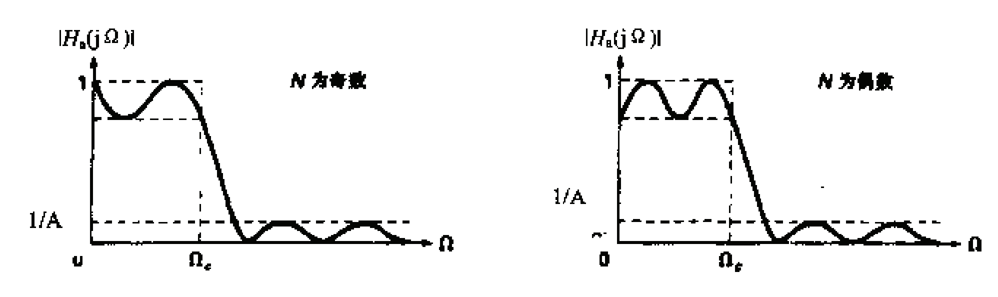


图 8.2.9　椭圆滤波器的幅频特性

其中：$k=\dfrac{\Omega_p}{\Omega_c}$，$k_1=\dfrac{\varepsilon}{\sqrt{A^2-1}}$，而 $K(x)=\displaystyle\int_0^{\pi/2}\frac{\mathrm{d}\theta}{\sqrt{1-x^2\sin^2\theta}}$

是完全的椭圆积分，MATLAB 提供了 ellipke 函数对上述积分做数值计算，它会在计算 N 和设计椭圆滤波器的子程序中用到，而我们只要调用子程序，可以不去管它。

设计椭圆滤波器当然要用计算机协助。MATLAB 提供了一个函数 [z0,p0,k0]=ellipap(N,Rp,As)，用它设计一个阶数为 N，通带波动为 R_p，阻带衰减为 A_s 的归一化椭圆模拟原型滤波器，它在数组 z0 中返回零点，数组 p0 中返回极点，并且返回增益 k0。如果需要具有任意 Ω_c 的未归一化的椭圆滤波器，可把由归一化滤波器的数组 $p0$ 和 $z0$ 乘以 Ω_c，增益 $k0$ 乘以 Ω_c^{Nm}，其中 Nm 为极点数减去零点数，得到 p,z 和 k。同时 MATLAB 也提供了一个 ellipord 函数,它可以由给定的滤波器技术指标 Ω_p,R_p, Ω_s 和 A_s，求得滤波器的阶数 N 和边缘频率 Ω_c，其调用格式为

[N, OmegaC] = ellipord (OmegaP, OmegaS, Rp, As, 's')

和巴特沃斯滤波器和切比雪夫滤波器一样，把这个函数中得出的 N 作为 ellipap 函数的输入变元，把两个函数衔接起来使用。又因 Ω_c 也已得到，就可以完成椭圆型滤波器的原型设计问题，下面仍将以同一例题来说明。

例 8.2.5：按例 8.2.2 的指标，f_p=5 kHz，R_p=1 dB，f_s=12 kHz，A_s=30 dB，要求设计椭圆型低通滤波器来实现。

解：我们直接用 MATLAB 程序来解题，程序名为 hc825。它和前几个程序很相似。

```
OmegaP=2*pi*5000; OmegaS=2*pi*12000;                % 输入设计指标
Rp=1; As=30;
[N, OmegaC] = ellipord(OmegaP, OmegaS, Rp, As, 's') % 调用椭圆设计函数
[z0,p0,k0] = ellipap(N,Rp,As)                       % 调用椭圆原型函数,得到左半平面零极点
```

在得到 $p0$,$z0$,$k0$ 后，计算系统的传递函数归一化系数向量 $b0$, $a0$, 绘制频率特性与零极点分布的语句和 hc823 中的完全相同，此处就予以省略。

程序运行后，得到如图 8.2.10 所示的曲线及数字结果。

```
N = 3
OmegaC = 3.1416e+004
z0 =      0 - 1.9535i
          0 + 1.9535i
p0 =      -0.2053 - 0.9870i
          -0.2053 + 0.9870i
         -0.5596
k0 =      0.1490
b0 =      1.0000       0       3.8160
a0 =      1.0000    0.9701    1.2460    0.5687
sos =     0        1.0000        0     1.0000      0.5596      0
```

```
          1.0000      0    3.8160   1.0000   0.4106   1.0162
   G = 1
```

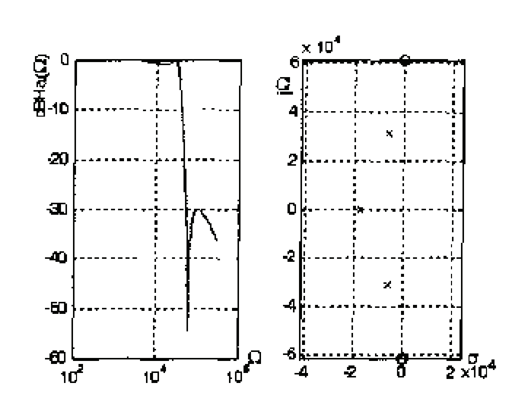


图 8.2.10　椭圆型模拟低通滤波器原型的对数幅频特性和零极点分布

所以，所设计的滤波器的传递函数为

直接型：$H_a(s)=\dfrac{s0^2+3.816}{s0^3+0.9701s0^2+1.2460s0+0.5687}$

级联型：

$$H_a(s)=\frac{s0(s0^2+3.816)}{s0(s0+0.3976)(s0^2+0.4106s0+1.0162)}=\frac{s0^2+3.816}{(s0+0.3976)(s0^2+0.4106s0+1.0162)}$$

其中 $s0=\dfrac{s}{31416}$ 为归一化拉普拉斯算子。

从图 8.2.10 得知，椭圆滤波器在通带和阻带内的幅特性都是等波动的，它充分利用了性能指标中的裕度。因此用这个方法设计出的滤波器只要三阶，就达到了其他滤波器用四阶或五阶才得到的指标，从而简化了系统结构和提高了效率。不过这只是从滤波器幅特性的角度讨论的结论。另外也可以看到，椭圆滤波器也兼有零点和极点，它的零点也在虚轴上。

8.2.5　滤波器原型的相位响应

椭圆滤波器提供了最优的平方幅度响应，但通带上的相位响应非线性较大（这是许多实际应用不希望的）。尽管在设计过程中不考虑相位响应，但相位在某些情况下是个重要的问题。巴特沃斯滤波器的幅度响应是单调的，为了得到相同的阻带指标，它需要更高的阶数（更多的极点）。然而它在通带上具有相当线性的相位响应。切比雪夫滤波器的相位特征介于两者之间。因此，在实际应用中，巴特沃斯滤波器、切比雪夫滤波器和椭圆滤波器都各有其适用的场合。选择哪种类型则取决于实际问题对滤波器的阶数（它影响处理的速度和实现的复杂性）和相位特性（它控制失真）的要求。

8.3　由模拟滤波器变换为数字滤波器

利用模拟滤波器的设计结果来求相应的数字滤波器，可以用映射的方法来完成。就是要把 s 平面映射到 z 平面，使模拟系统函数 $H_a(s)$ 变换成等价的数字滤波器的系统函数 $H(z)$，这种由复变量 s 到复变量 z 之间的映射（变换）关系，必须满足三条基本要求：

第一，$H(z)$的频率响应要能模仿 $H_a(s)$ 的频率响应，即 s 平面的虚轴必须映射到 z 平面的

单位圆上，也就是频率轴要对应。

第二，因果稳定的 $H_a(s)$ 应能映射成因果稳定的 $H(z)$。也就是 s 平面的左半平面 $\sigma<0$ 必须映射到 z 平面单位圆的内部 $|z|<1$。

第三，变换前后的滤波器在时域或频域的主要特征（频率响应或脉冲响应等）应尽可能相同或接近。

从模拟滤波器映射成数字滤波器也就是使数字滤波器能模仿模拟滤波器的特性，工程中通常采用双线性变换法。理论上说，还有脉冲响应不变法、阶跃响应不变法等，但很少应用。本书将着重讨论双线性变换法，也将介绍脉冲响应不变法作为比较，以便加深理解。

8.3.1 脉冲响应不变法

脉冲响应不变法是使数字滤波器的单位脉冲响应序列 $h(n)$模仿模拟滤波器的冲激响应 $h_a(t)$。第 2.1 节中讨论过这两者之间量纲的不同，要使两者建立等价关系，应当使 $h(n)$等于模拟滤波器的冲激响应的抽样值 $h_a(nT)$ 与采样周期 T 的乘积，即满足

$$h(n)=T\cdot h_a(nT) \tag{8.3.1}$$

连续系统脉冲响应由 s 平面上的各个极点的分量 p_{ak} 组成，而离散系统脉冲响应由 z 平面上的各个极点的分量 p_{dk} 组成。（下标 a 和 d 分别表示模拟和数字域）。两者之间的映射关系为 $p_{dk}=e^{p_{ak}T}$。它是由我们熟知的拉普拉斯算子 s 和 z 变换算子之间的关系式决定的。

$$z=e^{sT} \tag{8.32}$$

及

$$s=\frac{1}{T}\ln z \tag{8.3.3}$$

这个复平面到复平面的复数映射图形如图 8.3.1 所示，s 平面上每一条宽度为 $2\pi/T$ 的横条都将映射到整个 z 平面上，每一横条的左半边映射到 z 平面单位圆以内，右半边映射到 z 平面单位圆以外，而 s 平面虚轴 jΩ映射到 z 平面单位圆上。虚轴上每一长为 $2\pi/T$ 的线段都映射到 z 平面单位圆上一周。s 平面每个横条都要映射到整个 z 平面上，所以产生了无数映射的重叠。这种多对一的非单值的映射关系，反映了(4.4.7)式所表述的数字频谱是模拟频谱的周期延拓叠加关系；也说明了混叠产生的原因；因为通过这个变换，模拟域 s 平面上的许多频谱模态会映射到 z 平面的同一点上。

图 8.3.2 给出了这个映射式的详细内容。左边两个子图为 s 平面，右边为 z 平面。左上图为 s 平面上的等实部线，它们映射到 z 平面上成为以原点为中心的同心圆。s 平面上 $\sigma=0$ 的虚轴则映射到 z 平面的单位圆圆周上。图中只表示了 $\sigma<0$ 的左半 s 平面，因此它们全部映射到单位圆的内部。我们有意把这些垂直线的长度取为从-3~3，比 $-\pi\sim\pi$ 小一些，其相应的 z 平面上的圆就缺了一个角。所以，若把这些垂直线的长度向正负两方向无限延长的话，z 平面上的映射将在同心圆上不断重复转圈，反映出多点对一点的非单值映射。左下图是 s 平面上的等虚部线，也即是等 Ω 线，它们在 z 平面上的映射是等 ω 相角的径向直线，如右下子图所示。这说明模拟域中的频率 Ω，是以线性关系 $\omega=\Omega T$ 映射到数字域的。

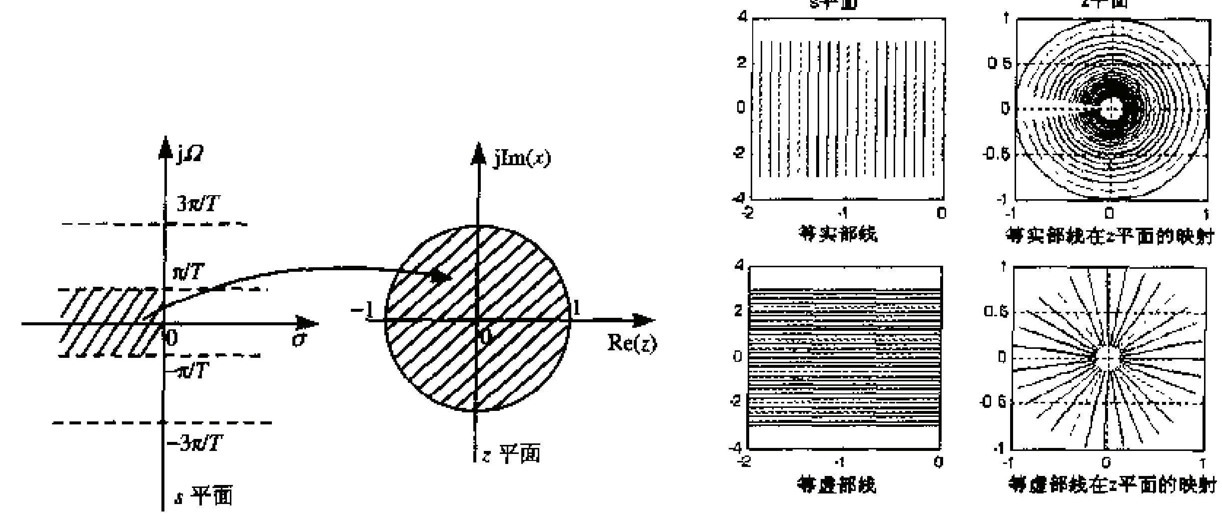


图 8.3.1　脉冲响应不变法的映射关系

图 8.3.2　脉冲响应不变法映射关系的内部细节

脉冲响应不变法是一种时域响应等价的方法。它以变换前后的模拟和数字滤波器脉冲响应等价性为基础，得到在这个意义下与模拟滤波器相应的数字滤波器。这种等价映射的思路如下：

（1）根据已知的模拟滤波器传递函数 $H_a(s)$，用拉普拉斯反变换求出它的冲激响应 $h_a(t)$；

（2）对此冲激响应进行采样，再乘以 T，得到等价的脉冲响应序列 $h_e(n)=T\cdot h_a(t)|_{t=nT}$；

（3）对 $h_e(n)$ 求 z 变换，得到其数字系统的传递函数 $H_e(z)$。

因此，用脉冲响应不变法来设计低通数字滤波器的过程是：首先按给定的数字低通滤波器技术指标 ω_p，ω_s，R_p，和 A_s，设计一个等价的模拟滤波器，然后把它映射成所期望的数字滤波器。具体的设计步骤为：

（1）首先按给定的指标 ω_p，ω_s，确定采样周期 T 并选择模拟频率。

$$\Omega_p=\frac{\omega_p}{T}\text{和}\Omega_s=\frac{\omega_s}{T} \tag{8.3.4}$$

（2）根据指标 Ω_p，Ω_s，R_p，和 A_s，设计模拟滤波器 $H_a(s)$。这可由上一节的四种原型（巴特沃斯、切比雪夫 I、切比雪夫 II、椭圆）滤波器之一来实现。

（3）利用部分分式展开式，把 $H_a(s)$ 写成

$$H_a(s)=\sum_{k=1}^{N}\frac{R_k}{s-p_{ak}} \tag{8.3.5}$$

因此，模拟滤波器的冲激响应为

$$h_a(t)=L^{-1}\left[H_a(s)\right]=\sum_{k=1}^{N}R_k e^{p_{ak}t}, \tag{8.3.6}$$

（4）对此冲激响应进行采样，并乘以 T，得到等价的脉冲响应序列

$$h_e(n)=T\cdot h_a(t)|_{t=nT}=T\cdot\sum_{k=1}^{N}R_k\left(e^{p_{ak}T}\right)^n=T\cdot\sum_{k=1}^{N}R_k p_{dk}{}^n \tag{8.3.7}$$

（5）对 $h_e(n)$ 求 z 变换，因为(8.3.7)式求和号中的每一个单项都是 n 的幂次函数，很容易查表得知它对应于单个数字极点 $p_{dk}=e^{p_{ak}T}$，由此得到等价的数字滤波器的传递函数

$H_e(z)$。

$$H_e(z)=T\cdot\sum_{k=1}^{N}\frac{R_k}{1-p_{dk}z^{-1}}=\sum_{k=1}^{N}\frac{T\cdot R_k}{1-e^{p_{ak}T}z^{-1}} \tag{8.3.8}$$

这个数字传递函数$H_e(z)$和原来的模拟传递函数$H_a(s)$在脉冲响应方面是等价的，那么在频率特性方面是否等价呢？对(8.3.7)两端做傅里叶变换，利用(4.4.9)式，得到等价数字滤波器的频率响应和原来模拟滤波器的频率响应间的关系为：

$$H_e(e^{j\omega})=H_e\left(e^{j\Omega T}\right)=\sum_{k=-\infty}^{\infty}H_a(j\Omega-jk\Omega_s) \tag{8.3.9}$$

可见，等价数字滤波器的频率响应是模拟滤波器频率响应的周期延拓。正如采样定理所讨论的，只有当模拟滤波器的频率响应是限带的，且频带限于折叠频率以内时，即

$$H_a(j\Omega)=0,\qquad |\Omega|\geqslant\frac{\pi}{T}=\frac{\Omega_s}{2} \tag{8.3.10}$$

才能使等价数字滤波器的频率响应重现滤波器频率响应，避免混叠现象的产生。此时有

$$H_e(e^{j\omega})=H_a(j\Omega)=H_a(j\omega/T)\qquad |\omega|\leqslant\pi \tag{8.3.11}$$

但是，任何一个实际的模拟滤波器频率响应都不可能是严格限带的，这就会产生周期延拓分量的频谱交叠，产生混叠失真。模拟滤波器的频率响应在折叠频率以上频段内衰减越大、越接近于限带系统，频谱混叠失真就越小。下面的例子将从数值上说明这一点。

在这里，把数字系统传递函数H加以下标e，表示等价。以区别于对模拟信号采样所得数字序列的z变换。在第2.1节中曾强调采样序列与原模拟信号是不等价的，必须把模拟信号乘以T，才有等价的前提。所以这个H_e和以前所用的H是不同的两个概念，两者的差别就是一个乘子T，所以在公式中所用的符号必须有所区别。在本章的IIR滤波器设计例和其他部分，都是用等价数字滤波器的概念，不会混淆，所以就把下标e略去了，请读者注意。

例 8.3.1：利用脉冲响应不变法，把$H_a(s)=\dfrac{s+1}{s^2+5s+6}$转换成等价的数字滤波器$H(z)$，其中$T$=0.1s。

解：首先，用部分分式展开$H_a(s)$：

$$H_a(s)=\frac{s+1}{s^2+5s+6}=\frac{2}{s+3}-\frac{1}{s+2}$$

极点为$p_1=-3$和$p_2=-2$，根据式（8.3.4）和T=0.1，得：

$$H_e(z)=\frac{2T}{1-e^{-3T}z^{-1}}-\frac{T}{1-e^{-2T}z^{-1}}=\frac{0.1-0.08966z^{-1}}{1-1.5595z^{-1}+0.6065z^{-2}}$$

当系统的阶次比较高时，手工计算是很麻烦的。需要开发一个MATLAB函数来实现脉冲响应不变映射。已知有理函数$H_a(s)$的分子分母系数向量ba和aa（变量中第二个字符a或d表示模拟域或数字域），先利用residue函数得到它的极点留数Ra,pa以及直接项Ca；然后根据（8.3.2）式把每一个模拟极点映射成数字极点$pd_k=e^{pa_kT}$。得到$H(z)$的极点留数部分分式；最后，用函数residuez把$H(z)$转换成传递函数形式。注意residue和residuez分别为模拟域和数字域中的极点留数函数。它们具有双向的功能，其变换的方向由输入变元的数目来判定。当输入变元为两个时，是由系数向量转为极点留数；输入变元为三个时，是由极点留数转为系数向量。因此脉冲响应不变映射法可以简洁地用下列MATLAB核心语句表示：

```
function [bd,ad]=impinvar0(ba,aa,Fs)
[Ra,pa, Ca]=residue(ba, aa);            % 将模拟滤波器系数向量变为模拟极点和留数
T=1/Fs; pd=exp(pa*T);                   % 将模拟极点变为数字（z 平面）极点pd
[bd,ad]=residuez(T*Ra, pd, Ca)          %用原留数Ra和数字极点pd求得数字滤波器系数向量
bd=real(bd), ad=real(ad)                %去除运算误差所产生的微小虚数
```

第一行语句把模拟域的系数行向量 *ba* 和 *aa* 转变为留数 *Ra*、极点 *pa* 和直接项 *Ca*，这三个输出变元均为列向量。对于 *aa* 向量比 *ba* 向量长度大的物理可实现情况，*Ca* 实际上是空矩阵。第三行语句的输入变元为三个，residuez 认为它们是数字滤波器的留数 *Rd*、极点 *pd* 和直接项 *Cd* 矩阵，并把它们转变为系数向量 *bd* 和 *ad*。要注意，即使 *Cd* 是空矩阵，仍必须把它放在输入变元中，凑足三个变元。否则 residue 将朝错误的方向变换，得不到正确的结果。还要强调的是第三行语句中取 Rd=T*Ra，这体现了(8.3.8)式，表明所求的是等价数字脉冲响应。

将例 8.3.1 的数据 b=[1,1]; a=[1,5,6]; T=0.1; 分别作为 ba,aa 输入 MATLAB 工作空间，再执行上面三句核心语句，就可以得到

```
bd =    1.0000   -0.8966
ad =    1.0000   -1.5595    0.6065
```

这与上面手工计算的结果一致。为了看出脉冲响应不变法可能造成的混叠现象，可以用下面的程序检验变换前后的滤波器的脉冲响应和幅频特性，并重叠在一张图上进行比较。

```
subplot(2,1,1), t=0:0.1:3;
ha=impulse(ba,aa,t);plot(t,ha*T),hold on   % 计算模拟系统的脉冲响应，乘以T
h=impz(bd,ad,31);stem(0.1*[0:30],h)        % 数字系统的脉冲响应，以同样时间轴画点列
subplot(2,1,2), w=[0:0.1:10]*2*pi;
Ha=freqs(ba,aa,w); plot(w,abs(Ha)),hold on % 计算模拟系统的频率特性，画曲线
H=freqz(bd,ad,w/Fs); plot(w/Fs,abs(H))     % 计算数字系统的频率特性，画曲线
plot(w,abs(H))                             % 将数字频率放大Fs倍，与模拟频率的幅特性比较
```

由于数字频率的取值范围为±π 或 0~2π，若要与模拟频率对比，就应当将数字频率乘以采样频率 F_s。所以在子图 8.3.3(b)中，把频率轴放大了 10 倍，以便比较两者的幅频特性。

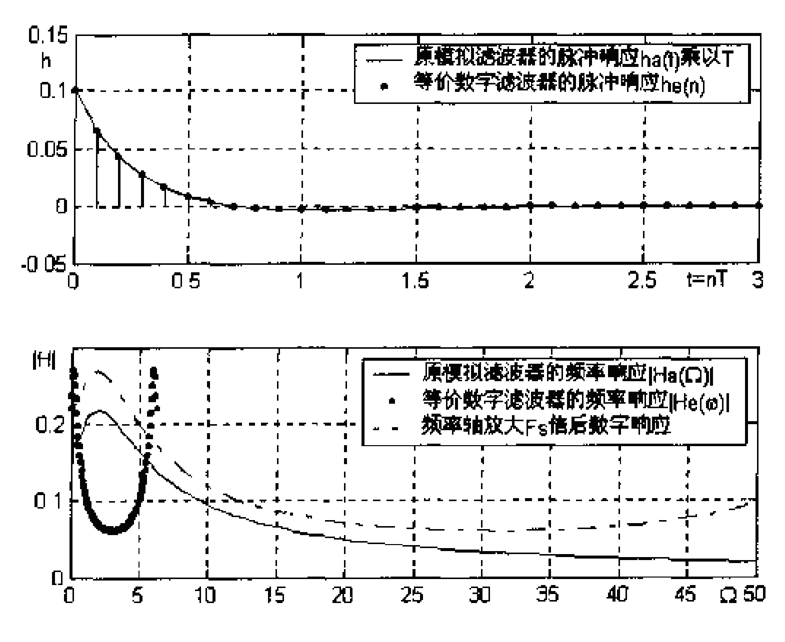


图 8.3.3　脉冲不变法变换前后的系统的时域和频域特性的比较

从图 8.3.3 可以看出，把模拟系统的脉冲响应采样并乘以 T 后与数字系统的脉冲响应是

重合的，而数字滤波器的幅频特性却高于变换前的模拟滤波器。本题中的采样频率 F_s 为 10 Hz，奈奎斯特频率为 31.4rad / s。根据采样定理，高于这个频率的模拟频谱将折叠回来叠加在主频谱上，所以在这个位置上，数字频率的幅特性约为模拟频率幅特性的两倍。而更高频区间的幅特性也会将折叠叠加进来，使数字滤波器的幅特性有所抬高。从图中可看到本例中模拟滤波器在 $F_s/2$ 处的幅特性约为 0.03，达到最大幅特性的 10%以上左右，所以混叠就比较严重。这是脉冲响应不变法的致命缺点，除非原来的模拟滤波器具有很好的低通滤波性能，才能使用。

MATLAB 信号处理工具箱中提供了一个名为 impinvar 的子程序，它用脉冲响应不变法来实现把模拟滤波器变换为数字滤波器。其调用格式为：[bd,ad] =impinvar (b,a,Fs)。由 impinvar 得到的系数向量 bd 和 ad 与本书所给的子程序 impinvar0 相同。因此，在例 8.3.1 中，可以键入[bd,ad] =impinvar ([1,1],[1,5,6],10)。

执行后将给出同样的答案。信号处理工具箱中的 impinvar 采用了状态方程编程，不易读懂。读者可以用本书提供的四条语句来理解脉冲响应不变法的编程思想。但调用时应该尽量用商品化的程序 impinvar，因为它考虑得周到，可以在错误输入时给出适当的提示。

下面用例子说明用脉冲响应不变法设计数字低通滤波器的全过程。

例 8.3.2：用巴特沃斯滤波器原型设计一个低通数字滤波器，满足：$\omega_p=0.2\pi, R_p = 1$ dB；$\omega_s=0.3\pi, A_s = 15$ dB，滤波器的采样频率为 1000 Hz。

解：用下面的 MATLAB 程序 hc832 将能说明设计的过程：

```
% 数字滤波器指标:
wp = 0.2*pi;  ws = 0.3*pi;                  % 数字通带和阻带频率(Hz)
Rp = 1;  As = 15;                           % 通带和阻带波动(dB)
% 转换为模拟原型指标
T = 0.001; Fs=1/T;                          % 置 Fs=1000
OmegaP = wp*Fs; OmegaS = ws*Fs;             % 原型通带频率和阻带频率
% 模拟巴特沃斯原型滤波器计算:
[N,OmegaC] =buttord(OmegaP,OmegaS,Rp,As,'s')  % 原型的阶数和边缘频率计算
[z0,p0,k0] = buttap(N);                     % 归一化巴特沃斯模拟原型设计
p = p0*OmegaC; z = z0*OmegaC;               % 将零极点乘以 Omegac，得到非归一化零极点
k = k0*OmegaC^N;                 % 将k0 乘以 Omegac^N,得到非归一化 k
ba =k* real(poly(z));                       % 由零点计算模拟滤波器分子系数向量
aa = real(poly(p))                          % 由极点计算模拟滤波器分母系数向量
[bd,ad] = impinvar(ba,aa,1/T);              % 调用脉冲响应不变法函数
[C,B,A]=tf2par(bd,ad)                       % 变换为并联二阶环节
```

此程序的运行结果为

```
OmegaC = 708.6537
N = 6
C = [ ]
B = 1.8701   -0.6294
    -2.1594  1.1475
    0.2893   -0.4503
A = 1.0000   -0.9918   0.2544
    1.0000   -1.0628   0.3671
    1.0000   -1.2898   0.6929
```

所设计的滤波器是一个六阶的巴特沃斯滤波器，其系统函数 $H(z)$的并联形式为：

$$H(z)=\frac{1.870-0.629z^{-1}}{1-0.992z^{-1}+0.254z^{-2}}+\frac{-2.159+1.148z^{-1}}{1-1.063z^{-1}+0.367z^{-2}}+\frac{0.289-0.450z^{-1}}{1-1.290z^{-1}+0.693z^{-2}} \qquad (8.3.12)$$

在图 8.3.4 中给出它的频率特性图。

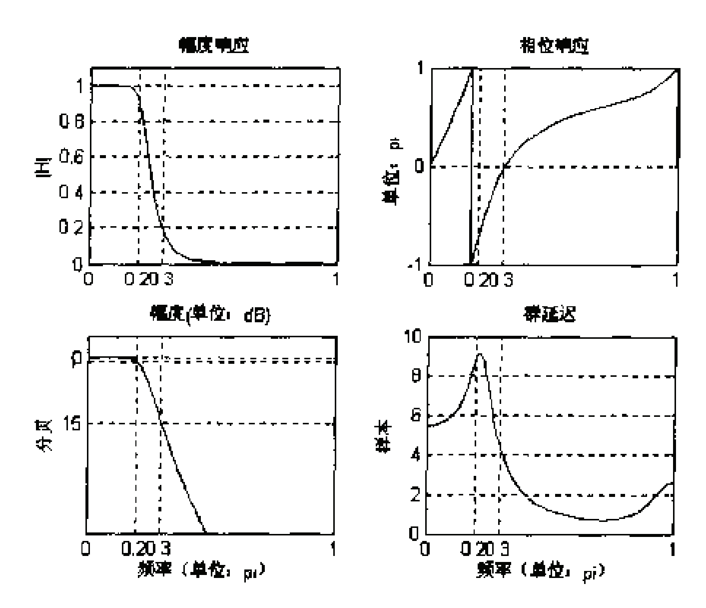


图 8.3.4　用脉冲响应不变法设计的巴特沃斯数字滤波器频率特性

从以上讨论看出，脉冲响应不变法使得数字滤波器的冲激响应完全模仿模拟滤波器的冲激响应。它的时域逼近良好，而且模拟频率 Ω 和数字频率 ω 之间呈线性关系。因而一个线性相位的模拟滤波器（例如贝塞尔滤波器）可以映射成一个线性相位的数字滤波器。但是，因为有频率混叠效应，所以脉冲响应不变法只适用于限带的模拟滤波器。高频区幅特性不等于零的高通和带阻滤波器不能采用脉冲响应不变法。对于带通和低通滤波器，要加抗混叠滤波器，滤掉高于折叠频率以上的频率。阻带衰减越大，则混叠效应越小。由于这许多限制和缺点，现在大多数情况都采用下节介绍的方法——双线性变换法。

8.3.2 双线性变换法

脉冲响应不变法是使数字滤波器在时域上模仿模拟滤波器，但是它们的缺点是频率响应产生的混叠，这是因为从 s 平面到 z 平面的多值映射关系所造成的。为了克服这一缺点，产生了双线性变换法。

双线性变换法是使数字滤波器的频率响应模仿模拟滤波器的频率响应的一种变换方法，它克服多值映射这一缺点。双线性变换法把整个 s 平面变换到整个 z 平面上去，而且也使 s 的左半平面映射到 z 平面的单位圆内。这样就使 s 平面与 z 平面是一一对应的关系，消除了多值变换性，也就消除了频谱混叠现象。它同时也保持了数字滤波器与原来的模拟滤波器有同样的稳定性和因果性，如图 8.3.5 所示。

这种映射的映射函数为：

$$s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\qquad\Rightarrow\qquad z=\frac{1+sT/2}{1-sT/2}\tag{8.3.13}$$

其中 T 是参数。通常取采样周期，也可以取其他值。这种变换的另一个名字叫做双线性分式变换，这是因为分式展开后，得到：

$$\frac{T}{2}sz+\frac{T}{2}s-z+1=0\tag{8.3.14}$$

若固定其中一个变量，则另一个是线性的。或者说，这种变换对于 s 和 z 是双线性的。

(8.3.14)的复平面映射如图 8.3.5 所示。图中左面两个子图是 s 平面等实部线和等虚部线，右面两个子图是它们在 z 平面上的映射曲线。可以看出，这些等实部线在 z 平面上的映射是一组圆：当它们在 s 平面上向 $\pm j\infty$ 延伸时，其在 z 平面上的映射都向(-1,0)点汇聚。其等虚部线在 z 平面上的映射是一组向外辐射的弧线。结果说明，双线性变换是把整个左半 s 平面（而不是一个条形区）一一对应地映射到单位圆的内部。具体的数学分析如下：

（1）把 $s=\sigma+j\Omega$ 代入（8.3.11）式，得到

$$z=\left(1+\frac{\sigma T}{2}+j\frac{\Omega T}{2}\right)\Big/\left(1-\frac{\sigma T}{2}-j\frac{\Omega T}{2}\right) \tag{8.3.15}$$

因此

$$\sigma<0 \;\Rightarrow\; |z|=\left|\frac{1+\frac{\sigma T}{2}+j\frac{\Omega T}{2}}{1-\frac{\sigma T}{2}-j\frac{\Omega T}{2}}\right|<1$$

$$\sigma=0 \;\Rightarrow\; |z|=\left|\frac{1+j\frac{\Omega T}{2}}{1-j\frac{\Omega T}{2}}\right|=1$$

$$\sigma>0 \;\Rightarrow\; |z|=\left|\frac{1+\frac{\sigma T}{2}+j\frac{\Omega T}{2}}{1-\frac{\sigma T}{2}-j\frac{\Omega T}{2}}\right|>1\text{。}$$

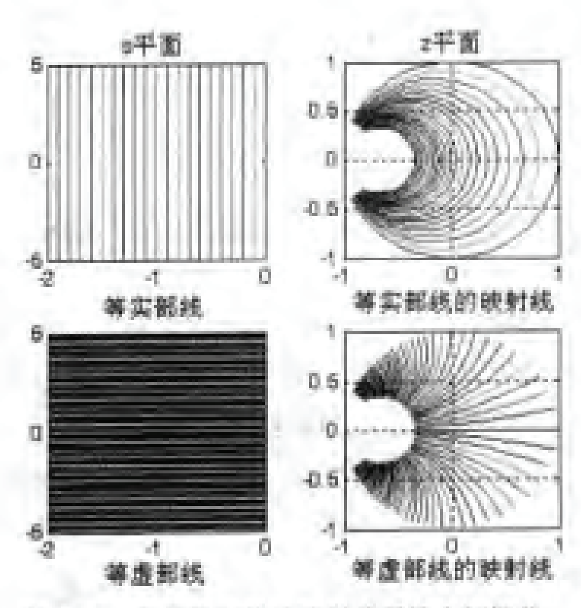


图 8.3.5　双线性变换法映射关系的内部细节

（2）整个左半平面映射成单位圆的内部，因此这是一个稳定的变换。

（3）虚轴以一一对应的方式映射成单位圆，因此在频域中不会出现混叠。

把 $\sigma=0$ 代入式（8.3.15），得到

$$z=\frac{1+\mathrm{j}\frac{\Omega T}{2}}{1-\mathrm{j}\frac{\Omega T}{2}}=\mathrm{e}^{\mathrm{j}\omega} \tag{8.3.16}$$

由于幅度为 1，把 ω 看做 Ω 的函数求解，得到

$$\omega=2\tan^{-1}\left(\frac{\Omega T}{2}\right) \tag{8.3.17 a}$$

$$\Omega=\frac{2}{T}\tan\left(\frac{\omega}{2}\right) \tag{8.3.17 b}$$

这说明 Ω 和 ω 的关系是非线性的，但是没出现混叠。在把 Ω 变换为 ω 时产生了非线性的崎变，为了克服它带来的问题，通常要使 Ω 按（8.3.17 b）预修正，以抵消崎变的影响。

（4）根据公式(8.3.9)和(8.3.11)所描述的模拟滤波器和经过采样后的数字滤波器之间的关系，可以写出：

$$H_{\mathrm{e}}(z)=H_{\mathrm{a}}(s)\Big|_{s=\frac{2}{T}\frac{1+z^{-1}}{1-z^{-1}}} \tag{8.3.18}$$

H 的下标 e 表示等价，由于在 IIR 滤波器设计中，$H(z)$都是等价变换来的，不会误解，以后将把下标 e 省略。下面着重看看如何对 $H_{\mathrm{a}}(s)$ 进行 $s=\frac{2}{T}\cdot\frac{1+z^{-1}}{1-z^{-1}}$ 的双线性变换。

例 8.3.3：利用双线性变换，把 $H_{\mathrm{a}}(s)=\frac{s+1}{s^2+5s+6}$ 转换成数字滤波器，设 $T=1$。

解：根据公式（8.3.18），得到

$$H_{\mathrm{a}}(s)=H_{\mathrm{a}}\left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\Bigg|_{T=1}\right)=H_{\mathrm{a}}\left(2\frac{1-z^{-1}}{1+z^{-1}}\right)=\frac{2\frac{1-z^{-1}}{1+z^{-1}}+1}{\left(2\frac{1-z^{-1}}{1+z^{-1}}\right)^2+5\left(2\frac{1-z^{-1}}{1+z^{-1}}\right)+6}$$

化简为：

$$H(z)=\frac{3+2z^{-1}-z^{-2}}{20+4z^{-1}}=\frac{0.15+0.1z^{-1}-0.05z^{-2}}{1+0.2z^{-1}}$$

这个例题中 $H_{\mathrm{a}}(s)$ 不过两阶，用手工化简已经够麻烦了。如果阶次稍高一些，手工计算是不可取的，肯定需要用 MATLAB 程序，而编程首先要掌握计算的规律。假如 $H_{\mathrm{a}}(s)$ 为 N 阶，它的分子分母都按 s 正幂降序排列，最后以常数项结束。

$$H_{\mathrm{a}}(s)=\frac{ba_1s^M+ba_2s^{M-1}+\cdots+ba_Ms+ba_{M+1}}{aa_1s^N+aa_2s^{N-1}+\cdots+aa_0s+ba_{N+1}}$$

将 $s=\frac{2F_s(1-z^{-1})}{1+z^{-1}}=\frac{Nz}{Dz}$ 代入并进行通分，得到下式（注意，为了简化公式，Nz 和 Dz 为用双线性变换的分子分母系数多项式，用 MATLAB 表示时 Nz=[2,-2], Dz=[1,1]）。

$$H(z)=H_a(s)\Big|_{s=\frac{2F_s(1-z^{-1})}{1+z^{-1}}=\frac{Nz}{Dz}}=\frac{ba_1\left(\dfrac{Nz}{Dz}\right)^M+ba_2\left(\dfrac{Nz}{Dz}\right)^{M-1}+\cdots+ba_M\left(\dfrac{Nz}{Dz}\right)+ba_{M+1}}{aa_1\left(\dfrac{Nz}{Dz}\right)^N+aa_2\left(\dfrac{Nz}{Dz}\right)^{N-1}+\cdots+aa_0\left(\dfrac{Nz}{Dz}\right)+ba_{N+1}} \tag{8.3.19}$$

$$=\frac{ba_1Dz^{N-M}Nz^M+ba_2Dz^{N-M+1}Nz^{M-1}+\cdots+ba_MDz^{N-1}Nz+ba_{M+1}Dz^N}{aa_1Nz^N+aa_2DzNz^{N-1}+\cdots+aa_0Dz^{N-1}Nz+ba_{N+1}Dz^N}$$

可以看出如下的规律，分子分母中每一项都由常数乘以系数向量 Nz 和 Dz 的乘幂组成，它表示系数多项式的乘积，这乘积在 MATLAB 中由 conv 函数来完成，即

Nz^2=conv(Nz,Nz), Dz^2=conv(Dz,Dz), NzDz=conv(Nz,Dz)…

式(8.3.19)中各项的 Nz 和 Dz 的幂次和均为 N。N 次乘幂要由调用 N-1 次 conv 来完成。

为了便于同样的程序进行计算，把分子分母取成同样长度，在一般情况下 $M \leqslant N$，所以用在 ba 左边补零的方法把和 aa 调整成同样长度 N+1。此时就可以按(8.3.19)式中的分母形式来编程，计算分母的多项式系数 ad。这个程序也适合分子，只要代入数组 ba，可得出 bd。

下面就是自编的实现双线性变换的子程序 bilinear0.m。

```
function [bd,ad] = bilinear0(ba,aa,Fs)
% 从s域到z域的频带变换
% 将ba,aa中的短者左边补零成同长
lba=length(ba);laa=length(aa); ld=laa-lba; % 求分母与分子阶次之差ld
if ld>=0
ba=[zeros(1,ld),ba];                        % 若aa长度大于ba,给ba前补ld个零
else aa=[zeros(1,-ld),aa];                  % 若ba长度大于aa,给aa前补ld个零
end
Nz = 2*Fs* [1,-1]; Dz=[1,1];                % 双线性变换分子分母系数向量
N = max(lba,laa)-1;                         % 模拟系统阶次N
bd = 0; ad = 0;                             % bd,ad系数向量初始化
for k = 0:N
   pld = [1];pln = [1];                     % 双线性变换分子分母系数乘积项初始化
   for l = 0:k-1
      pld = conv(pld,Dz);                   % 求双线性变换分母系数k次幂Dz^k
   end
   for l = 0:N-k-1
      pln = conv(pln,Nz);                   % 求双线性变换分子系数(N-k)次幂Nz^(N-k)
   end
   bd = bd+ba(k+1)*conv(pln,pld);           % 分子系数多项式向量求和
   ad = ad+aa(k+1)*conv(pln,pld);           % 分母系数多项式向量求和
end
ad1 = ad(1); ad = ad/ad1; bd = bd/ad1;      % 用分母系数多项式的首项使分子分母系数归一化
```

信号处理工具箱也提供了一个叫做 bilinear 的函数实现这种映射，它的调用方法与 impinvar 函数相似，可用 help 语句查阅。在下面的例子中将说明它的用法。

例 8.3.4：用 bilinear 函数对例 8.3.3 中的系统函数 $H_a(s)$进行变换，但把 T 改为 0.1。这时用手工计算就更烦琐一些了。

解：MATLAB 程序 hc834：

```
ba=[1,1]; aa=[1,5,6]; T=0.1; Fs=1/T;
[bd,ad] = bilinear(ba,aa,Fs);
bd =   0.0415    0.0040   -0.0375
ad =   1.0000   -1.5573    0.6047
```

滤波器传递函数为 $H(z)=\dfrac{0.0415+0.0040z^{-1}-0.0375z^{-2}}{1-1.5573z^{-1}+0.6047z^{-2}}$。

也可以计算出原模拟滤波器的脉冲响应和频率特性，并与变换为数字滤波器后的脉冲响应和频率特性进行比较。注意在进行脉冲响应的比较时，同样要把模拟脉冲响应 $ha(t)$ 的采样值乘以采样周期 T，才能与数字脉冲响应作等价性的比较。为此编程如下：

```
subplot(2,1,1),t=0:0.1:3;
ha=impulse(ba,aa,t);plot(t,ha*T),hold on   % 计算模拟系统的脉冲响应乘以T
h=impz(bd,ad,31);stem(0.1*[0:30],h,'.')     % 数字系统的脉冲响应，以同样时间轴画点列
subplot(2,1,2), w=[0:0.1:10]*2*pi;
Ha=freqs(ba,aa,w); plot(w,abs(Ha)),hold on% 计算模拟系统的频率特性，画曲线
H=freqz(bd,ad,w/Fs); plot(w/Fs,abs(H))      % 计算数字系统的频率特性，画曲线
plot(w,abs(H))        % 将数字系统的频率特性横轴放大Fs倍，便于和模拟幅特性比较
```

程序运行的结果见图 8.3.6。可以看出，两者的脉冲响应在起始点重合得不好，而在以后的各点还是很相符的。两者的幅特性在低频区域形状和峰值大小很相似，只是它们的峰值频率发生了变化。把数字频率除以 T，在此处是放大 10 倍，它的低频段和模拟幅特性几乎重合，而到高频区就差别很大了。数字频率在 π 处的幅特性为零，这说明频率混叠的现象得到了消除。把图 8.3.6 与图 8.3.4 做一比较，容易看出，双线性变换比脉冲响应不变法有很大的优越性，所以它得到了广泛的应用。

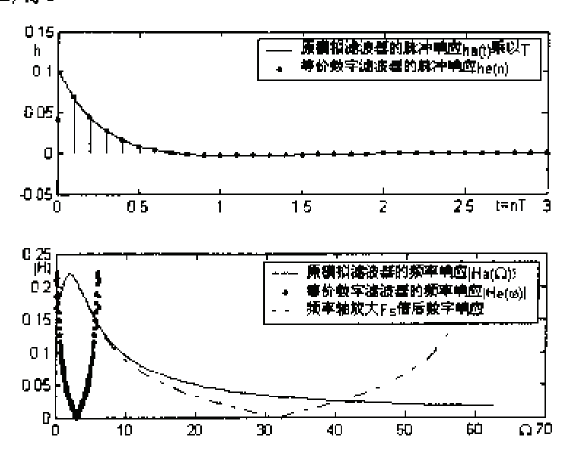


图 8.3.6　双线性变换前后的系统的时域和频域特性的比较

频率的非线性偏移的问题，是双线性变换中无法避免的一个特殊性质。道理很简单，数字频率的最大值为 π，而模拟频率可以向 ∞ 延伸，两者之间又要保持一一对应的映射关系（不像脉冲响应不变法那样的多值的映射）。所以两种频率之间就必然是非线性的关系。

(8.3.17)式表示了双线性变换中模拟角频率 Ω 与数字频率 ω 之间的关系，重写如下：

$$\Omega=\frac{2}{T}\tan\left(\frac{\omega}{2}\right) \tag{8.3.17}$$

它表明 s 平面与 z 平面是单值的一一对应关系，s 平面整个 Ω 轴单值对应于 z 平面单位圆的一周，即频率轴是单值变换关系。这个关系表示在图 8.3.7 上。

由于 s 平面的正、负虚轴 $\pm\mathrm{j}\Omega$ 分别映射成 z 平面单位圆的上半部分（辐角 ω 为正）及下半部分（辐角 ω 为负），频率轴是单值变换关系，且 $\Omega\to\infty$ 时，$\omega\to\pi$，即奈奎斯特折叠频率。故不会有高于折叠频率的分量，这就避免了脉冲响应不变法的频率响应混叠现象。在零频率附近，(8.3.17)式的频带变换关系还比较接近线性关系；当 ω 增加时，频率 Ω 与 ω 之间

出现了严重的非线性关系。

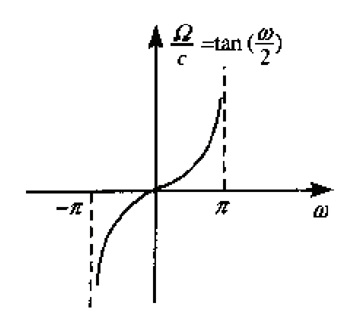


图 8.3.7　双线性变换中数字频率和模拟频率之间的关系

这种频率之间的非线性变换关系也带来了问题。首先，一个线性相位的模拟滤波器经双线性变换后就得到非线性相位的数字滤波器，不再保持原有的线性相位了；其次，对于幅频响应是分段常数型的模拟滤波器，虽然在双线性变换后，仍得到幅频特性为分段常数的滤波器，但是各分段边缘频率不能保持原来的比例关系。

这种频率的畸变，可通过频率的预修正加以校正，做法是将边缘频率事先加以修正，使它经变换后正好映射到所需频率。例如要求数字带通滤波器的四个边缘频率为 $\omega_1,\omega_2,\omega_3,\omega_4$，如果按线性变换，所对应的模拟滤波器的四个边缘频率应分别为：$\Omega_i=\omega_i T$，$(i=1,2,3,4)$ 。

但是用这四个频率的模拟滤波器经双线性变换 $\omega=2\tan^{-1}(\Omega T/2)$ 后，所得到的数字滤波器边缘频率显然就不等于原来要求的频率 $\omega_1,\omega_2,\omega_3,\omega_4$ 。因而要将频率加以预修正。做法是将这组数字频率 ω_i $(i=1,2,3,4)$ 按 $\Omega_i=\frac{2}{T}\tan\left(\frac{\omega_i}{2}\right)$ 的关系，变换成模拟频率 $\Omega_i(i=1,2,3,4)$，利用这组做过修正的模拟频率来设计模拟带通滤波器作为模拟原型。然后对此模拟原型滤波器采用双线性变换，变成数字滤波器。这时它的边缘频率应该正是我们原先所要求的那组 $\omega_1,\omega_2,\omega_3,\omega_4$。这一预修正过程可从图 8.3.8 看出。

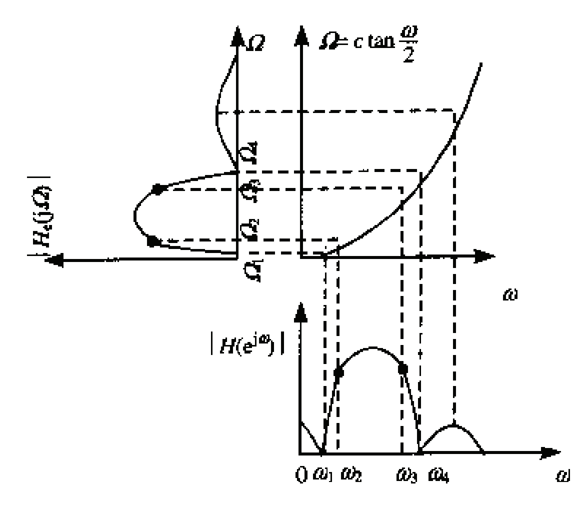


图 8.3.8　双线性变换的频率的预修正

由此可以归纳出采用双线性变换方法设计 IIR 数字低通滤波器的步骤。假如数字滤波器技术指标 ω_p, ω_s, R_p, 和 A_s 已经给定。

（1）选择 T 值，可以取实际系统中的值，也可以任意置为 1。在理论上它对变换的最后结果没有影响，但它会影响计算过程的精度。一般地说，使 $\Omega_c T$ 在 1 左右比较适当，不宜取太大或太小的数。

（2）对边缘频率 ω_p 和 ω_s 进行预修正，也就是根据（8.3.17 b）计算 Ω_p 和 Ω_s：

$$\Omega_p=\frac{2}{T}\tan\left(\frac{\omega_p}{2}\right),\qquad \Omega_s=\frac{2}{T}\tan\left(\frac{\omega_s}{2}\right)$$

（3）设计模拟滤波器 $H_a(s)$,使之满足技术指标 Ω_p, Ω_s,R_p 和 A_s，这在前一节中已讨论过。

（4）最后，令 $H(z)=H_a\left(\frac{2}{T}\cdot\frac{1-z^{-1}}{1+z^{-1}}\right)$，化简后得到为 z^{-1} 的有理函数的 $H(z)$。

在下面的几个例子中，将说明此设计过程。

例 8.3.5：设计例 8.3.2 的数字巴特沃斯滤波器，设滤波器的采样频率为 1000 Hz。.

解：MATLAB 程序 hc835

```
% 数字滤波器指标：
wp = 0.2*pi;  ws = 0.3*pi;                % 数字通带和阻带频率(Hz)
```

```
Rp = 1; As = 15;                              % 通带和阻带波动(dB)
% 转换为模拟原型指标
T = 0.001; Fs = 1/T;                          % 置 T=0.001
OmegaP = (2/T)*tan(wp/2);                     % 原型通带频率预修正
OmegaS = (2/T)*tan(ws/2);                     % 原型阻带频率预修正
% 模拟巴特沃斯原型滤波器计算:
[N,OmegaC] =buttord(OmegaP,OmegaS,Rp,As,'s');   % 原型的阶数和边缘频率计算
[z0,p0,k0] = buttap(N);                       % 归一化巴特沃斯原型设计函数
p = p0*OmegaC; z = z0*OmegaC;                 % 将零极点乘以 Omegac, 得到非归一化零极点
k = k0*OmegaC^N;                              % 此式只适用于无零点的情况
ba =k* real(poly(z));                         % 由零点计算分子系数向量
aa = real(poly(p))                            % 由极点计算分母系数向量
[bd,ad] = bilinear(ba,aa,Fs);                 % 双线性变换:
[sos,G] = tf2sos(bd,ad)                       % 变为二阶环节级联结构
```

程序运行的结果为

```
N =     6
OmegaC =  766.23
sos =    1.0000    2.0004    1.0000    1.0000   -0.9044    0.2155
         1.0000    2.0190    1.0194    1.0000   -1.0106    0.3583
         1.0000    1.9806    0.9810    1.0000   -1.2686    0.7051
G =      7.3782e-004
```

设计的滤波器是一个六阶的滤波器，由于 $H_a(s)$ 在 $s=-\infty$ 处有六个零点。这些零点在 z 平面的映射应该在 $z=-1$ 处。由于 MATLAB 的精度有限，这些零点不会精确地位于 $z=-1$ 处，表现在 sos 左三列的系数有误差。把它们调整到正确值后，系统函数应该为：

$$H(z)=\frac{0.7378\times10^{-4}\times(1+z^{-1})^6}{(1-0.9044z^{-1}+0.2155z^{-2})(1-1.0106z^{-1}+0.3583z^{-2})(1-1.2686z^{-1}+0.7051z^{-2})}$$

频率响应曲线如图 8.3.7 所示，可以看到它与图 8.3.4 非常相似。

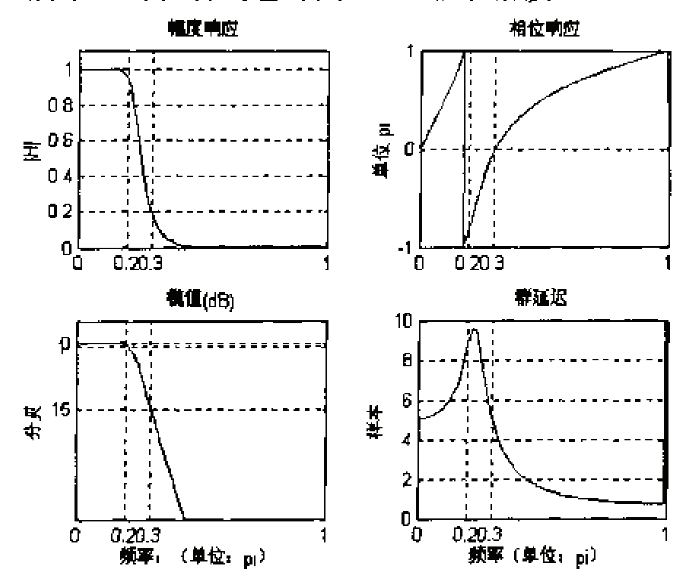


图 8.3.7　用双线性变换法设计的巴特沃斯数字滤波器频率特性

在上面的程序中，有一段是可以简化的。那就是不必求非归一化模拟滤波器系数 *ba* 和 *aa*，可以从归一化的系数 *ba*0 和 *aa*0 直接经双线性变换得到 *bd* 和 *ad*。只要把 bilinear 函数中的采样频率 *Fs* 用归一化的采样频率 Fs/OmegaC 代入。即

[bd,ad]=bilinear(ba0,aa0,Fs/OmegaC)

就可以求出 bd 和 ad。以后的程序中就用这个方法。

例 8.3.6：用切比雪夫 I 型滤波器作为原型设计数字滤波器，指标同例 8.3.5。

解：MATLAB 程序 hc836 如下：

```
% 滤波器指标变换和预修正同例 8.3.5，相关的语句从略
% 模拟切比雪夫 I 原型滤波器计算：
[N,OmegaC] = cheb1ord(OmegaP,OmegaS,Rp,As,'s');  % 原型的阶数和边缘频率计算
[z0,p0,k0] = cheb1ap(N,Rp);                  % 归一化切比雪夫 I 原型设计函数
ba0 = k0*real(poly(z0));                     % 由零点计算分子系数向量
aa0 = real(poly(p0))                         % 由极点计算分母系数向量
[bd,ad] = bilinear(ba0,aa0,Fs/OmegaC);  % 双线性变换：
[sos,G] = tf2sos(bd,ad)                      % 变为二阶环节级联结构
```

程序运行的结果为

```
N =   4
OmegaC =  649.8394
sos = 1.0000    2.0000    1.0000    1.0000   -1.5548    0.6493
      1.0000    2.0000    1.0000    1.0000   -1.4996    0.8482
G =   0.0018
```

设计的滤波器是一个四阶滤波器，在 $z=-1$ 处有四重零极点，系统函数为下式，频率响应图如图 8.3.9 所示。

$$H(z)=\frac{0.0018(1+z^{-1})^4}{(1-1.5548z^{-1}+0.6493z^{-2})(1-1.4996z^{-1}+0.8482z^{-2})}$$

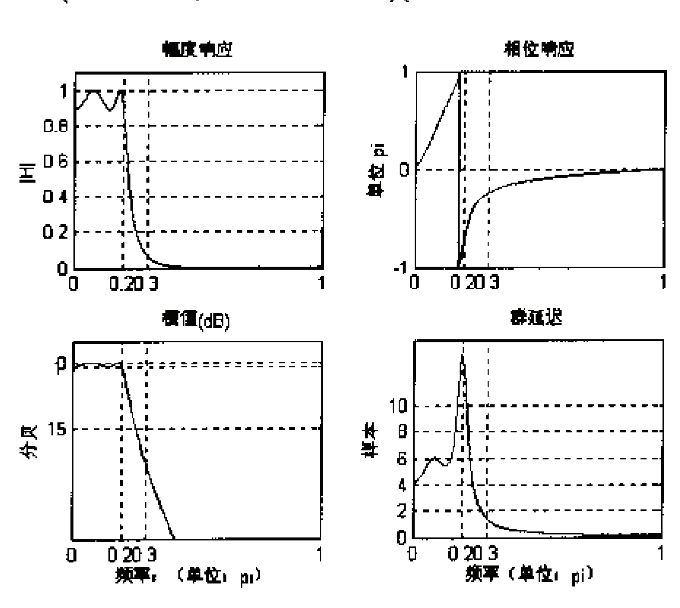


图 8.3.8 用双线性变换法设计的切比雪夫 I 型数字滤波器频率特性

例 8.3.7：用切比雪夫 II 型滤波器作为原型，设计数字滤波器，指标：同例 8.3.5。

解：MATLAB 程序 hc837 如下：

```
% 滤波器指标变换和修正：同例 8.3.5，程序语句从略
% 模拟切比雪夫 II 原型滤波器计算：
[N,OmegaC] = cheb2ord(OmegaP,OmegaS,Rp,As,'s');% 原型的阶数和边缘频率计算
[z0,p0,k0] = cheb2ap(N,As);                     % 归一化切比雪夫 II 原型设计函数
ba0 = k0*real(poly(z0));                        % 由零点计算分子系数向量
aa0 = real(poly(p0))                            % 由极点计算分母系数向量
[bd,ad] = bilinear(ba0,aa0,Fs/OmegaC);          % 双线性变换：
```

```
[sos,G] = tf2sos(bd,ad)                                    % 变为二阶环节级联结构
```

程序运行的结果为

```
N =   4
OmegaC =  851.85
sos =    1.0000    0.2133    1.0000    1.0000   -0.6039    0.1884
         1.0000   -1.2989    1.0000    1.0000   -1.3087    0.7476
G = 0.1653
```

设计的滤波器是一个四阶滤波器，频率特性见图 8.3.9，系统函数为

$$H(z)=\frac{0.1653(1+0.2133z^{-1}+z^{-2})(1-1.2989z^{-1}+z^{-2})}{(1-0.6039z^{-1}+0.1884z^{-2})(1-1.3087z^{-1}+0.7476z^{-2})}$$

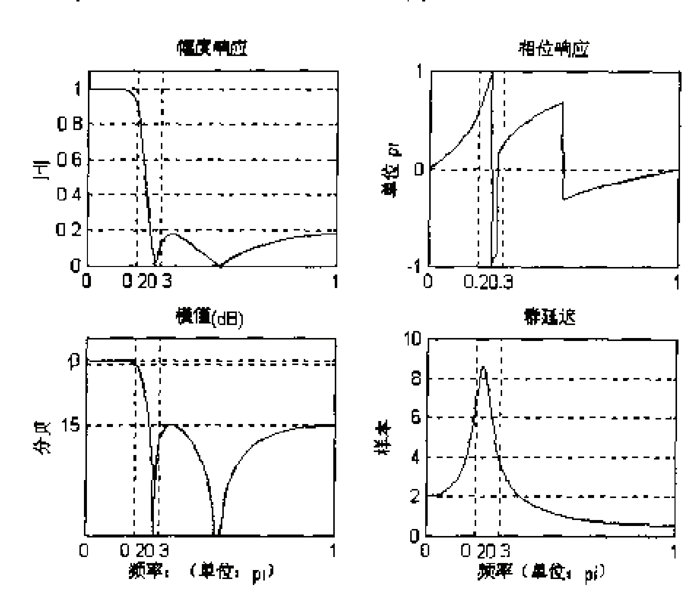


图 8.3.11　用双线性变换法设计的切贝雪夫 II 型数字滤波器频率特性

例 8.3.8： 用椭圆型滤波器作为原型，设计数字滤波器，指标同例 8.3.5。

解： MATLAB 程序 hc838 如下：

```
% 滤波器指标变换和予修正：同例 8.3.5，程序语句从略
% 模拟椭圆原型滤波器计算：
[N,OmegaC] = ellipord(OmegaP,OmegaS,Rp,As,'s');     % 原型的阶数和边缘频率计算
[z0,p0,k0] = ellipap(N,Rp,As);                      % 归一化椭圆原型设计函数
ba0 = k0*real(poly(z0));                            % 由零点计算分子系数向量
aa0 = real(poly(p0))                                % 由极点计算分母系数向量
[bd,ad] = bilinear(ba0,aa0,Fs/OmegaC);              % 双线性变换；
[sos,G] = tf2sos(bd,ad)                             % 变为二阶环节级联结构
```

程序运行的结果为

```
N =     3
OmegaC =  649.8394
sos =    1.0000    1.0000        0    1.0000   -0.6183       0
         1.0000   -1.4211    1.0000    1.0000   -1.4928    0.8612
G =    0.1214
```

设计的滤波器是一个三阶滤波器，频率特性见图 8.3.10，系统函数为

$$H(z)=\frac{121.4(1+z^{-1})(1-1.4211z^{-1}+z^{-2})}{(1-0.6183z^{-1})(1-1.4928z^{-1}+0.8612z^{-2})}$$

双线性变换最大的优点是避免了频率响应混叠现象，又能保证设计是稳定和因果的，而且对进行变换的滤波器类型没有限制。因此，这种方法广泛用在工程实践中。

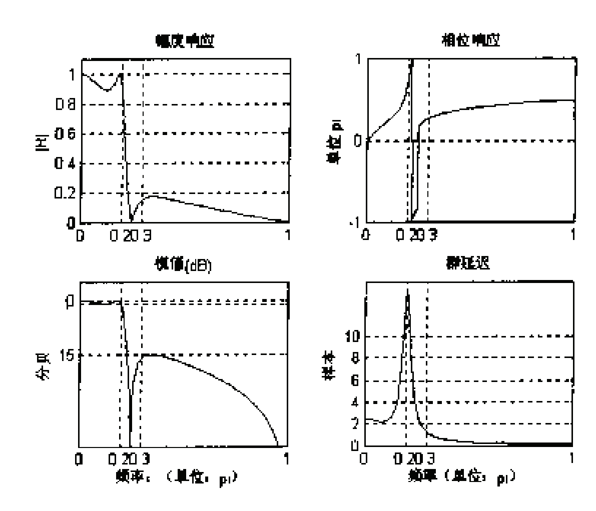


图 8.3.10　用双线性变换法设计的椭圆型数字滤波器频率特性

8.4　模拟滤波器的频带变换

本节讨论在模拟域把低通滤波器变换为高通、带通或带阻滤波器的方法，也就是图 8.1.3 中方法 1 中的 s→s 的变换。讨论的都是模拟滤波器，所以不在每处说明。

高通、带通和带阻滤波器的传递函数可以通过频带变换，分别由低通滤波器的传递函数求得。因此不论设计哪一种滤波器，都可以先将该滤波器的技术指标转换为低通滤波器的技术指标，按照该技术指标先设计低通滤波器。再通过频带变换，将低通的传递函数转换成所需类型的滤波器传递函数。

为了防止符号混淆，先做规定如下：原有低通滤波器传递函数用 $G(s0)$ 表示，所需变换后滤波器类型（例如高通）的传递函数用 $H(s)$ 表示考虑频率特性时，$s0=\mathrm{j}\Omega 0$，$s=\mathrm{j}\Omega$，$\Omega 0$ 表示变换前的频率，Ω表示变换后的频率。

8.4.1 低通到低通的频带变换

为了完整，把低通到另一低通的变换也放在这里归类。如果已经有了一个设计好的边缘频率为 $\Omega 0c=1$ 的归一化低通滤波器，现在要得到另一个边缘频率为 Ω_c，而其他指标相同的滤波器，那就可以不必从头设计，只要把原来的滤波器进行一个频带变换，使

$$s=\Omega_c\cdot s0 \tag{8.4.1}$$

即只要对原滤波器的频率都乘上一个常数，就实现了低通到低通的频带变换。其实我们在模拟滤波器设计中已经这样做过，先设计了归一化的滤波器，然后把它的拉普拉斯变量 $s0$ 换成 $s0=s/\Omega_c$，就得到了边缘频率为 Ω_c 的滤波器。概括来说，如果归一化低通传递函数为 $G(s0)$，则去归一化低通滤波器的传递函数 $H(s)$ 用下式转换：

$$H(s)=G(s0)\big|_{s0=s/\Omega_c} \tag{8.4.2}$$

这种乘以常数的频带变换，把 $\Omega0=0$ 变换到 $\Omega=0$，把 $\Omega0=\infty$ 变换到 $\Omega=\infty$，除了边缘频率 Ω_c 外，原来的滤波器指标都没有变化，低通还是低通。通带和阻带边缘频率都是成比例变化，所以容易理解和接受。下面要讨论的是把低通变成别的特性的滤波器，这时对频率轴要做的是非线性的变换，因而就需要绕一些弯子了。

8.4.2 低通到高通的频带变换

低通滤波器 $G(\mathrm{j}\Omega0)$ 和高通滤波器 $H(\mathrm{j}\Omega)$ 的幅频特性是反的。要由前者变换成后者，就要把通带从低频区换到高频区，把阻带由高频区换到低频区，如图 8.4.1 所示。不难想像，对原滤波器频率变量作一个倒数变换将能够实现这样的功能，而边缘频率则可用常数乘子来调整。令

$$\Omega = k/\Omega0 \tag{8.4.3}$$

其中 k 由给定的高通滤波器通带边缘频率 Ω_c 决定。将归一化滤波器的通带边缘频率 $\Omega0c=1$ 代入（8.4.3），要求左端为 Ω_c，由此得知 $k=\Omega_c$。于是 $s=\Omega_c/s0$。因而，低通到高通的变换为

$$H(s) = G(s0)\big|_{s0=\frac{\Omega_c}{s}} \tag{8.4.4}$$

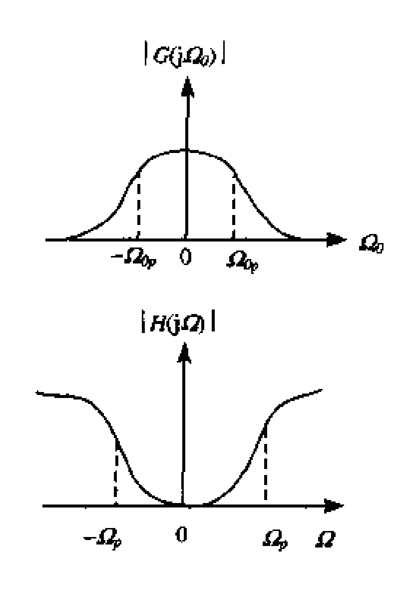
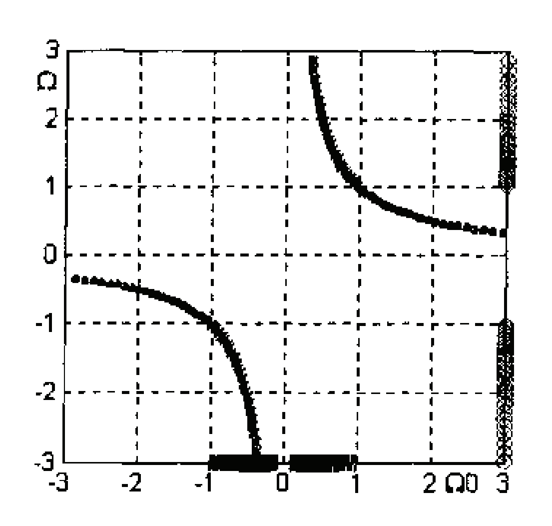


图 8.4.1 低通与高通的频率特性图



8.4.2 低通到高通的频率变换关系

设 $k=\Omega_c=1$，按(8.4.4)式画出的 $\Omega_0\sim\Omega$ 曲线见图 8.4.2。横坐标为归一化低通滤波器频率，纵坐标是变换后的高通滤波器频率。从-1～0～1 的低通通带变换为-1～-∞和∞～1 的两段高通频段。低通和高通的边缘频率 Ω_c 也用(8.4.3)式转换。由此得出模拟高通滤波器的设计步骤如下：

（1）确定高通滤波器的技术指标：通带边缘频率 Ω_c，通带波动 R_p，阻带衰减 A_s；

（2）按照上述 R_p，A_s 及通带边缘频率 $\Omega0_c=1$，设计归一化低通滤波器 $G(s0)$；

（3）求模拟高通的 $H(s)$。将 $G(s0)$ 按照(8.4.4)式，转换成高通滤波器 $H(s)$。

多数情况下，人们把 Ω_p 取为 Ω_c，有些情况下，特别关注的是带阻边缘频率 Ω_s，就取 $\Omega_c=\Omega_s$；也有些情况下，取通带和阻带边缘频率的几何平均值 $\Omega_c=\sqrt{\Omega_p \cdot \Omega_s}$ 作为边缘频率。这些都只影响特征频率 Ω_c 的取法，设计的基本步骤不变。

8.4.3　低通到带通的频带变换

归一化低通与带通滤波器的幅度特性如图8.4.3所示。图中 Ω_{c1} 和 Ω_{c2} 分别是带通滤波器的通带下边频和上边频边缘频率。如前面所说，它根据实际情况可以取 Ω_p， Ω_s 或它们的几何平均值。带通滤波器一般用通带中心频率 $\pm\Omega_z$ 和通带带宽 B 两个参数来表征。B 通常为作归一化的基准，因为它和低频滤波器的通带边缘频率意义相当。这两个参数的计算式如下

$$B=\Omega_{c2}-\Omega_{c1} \tag{8.4.5}$$

$$\Omega_z^2=\Omega_{c1}\Omega_{c2} \tag{8.4.6}$$

现在需要寻找一种变换，使得低通滤波器频率轴上 $\Omega0=0$ 的点变为带通滤波器的两个中心频率 $\pm\Omega z$，而 $\Omega0=\pm1$ 的点分别变为带通滤波器正负两个通带的上下边频 Ω_{c1} 和 Ω_{c2}，并保证其带宽为 B。

根据以上分析，这个变换应当把低通滤波器的全频带转换为带通滤波器的正半频率区间。由于带通滤波器频率特性是对称的，它的负半频率区间也必须由低通滤波器映射而成。因此这一个变换必须是一对二的多值映射，可以想像它应该是 Ω 的二次函数，在开方时取正取负都能满足。

研究说明，下列二次函数的映射公式(8.4.7 a)、(8.4.7 b)能满足要求。

$$\Omega0=\frac{\Omega^2-\Omega_z^2}{B\Omega} \tag{8.4.7 a}$$

其逆函数为

$$\Omega=\frac{-B\Omega0\pm\sqrt{B^2\Omega0^2+4\Omega_z^2}}{2} \tag{8.4.7 b}$$

在 B=1 时， 按这个函数画出的 $\Omega0\sim\Omega$ 曲线如图 8.4.4 所示。它表明了这个函数如何把低通频带变换为带通频率。这函数由上下两段曲线组成，每一段曲线映射出正负半轴上的一个通带。其中心频率为 Ωz,带宽为 B。将这个关系转到复频域，把 $s0=\mathrm{j}\,\Omega0$ 及 $s=\mathrm{j}\,\Omega$ 代入，有

$$s0=\frac{s^2+\Omega_z^2}{B\cdot s} \tag{8.4.8}$$

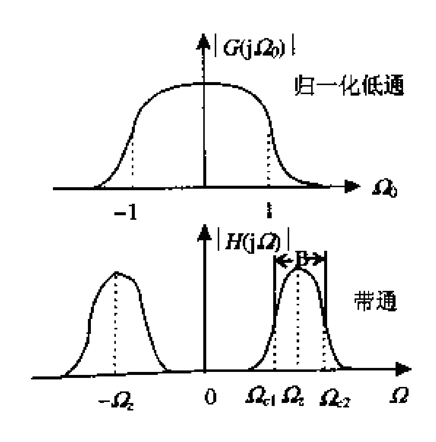


图 8.4.3　带通和低通滤波器的频率特性对应关系

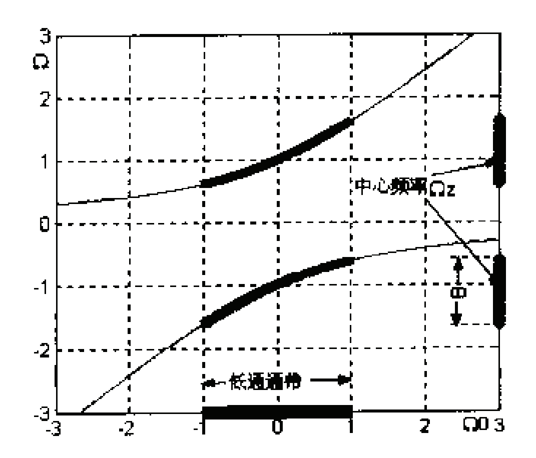


图 8.4.4　带通和低通滤波器的频率映射关系

(8.4.7)式是低通到带通的频带变换公式。利用该式将带通的边缘频率转换成低通的边缘频率。因此两者的传递函数之间满足如下关系

$$H(s)=G(s0)\Big|_{s0=\frac{s^2+\Omega_z^2}{B\cdot s}} \tag{8.4.9}$$

(8.4.9)式就是由归一化低通直接转换成带通的计算公式。因此，模拟带通滤波器的设计步骤如下：

（1）确定模拟带通滤波器的技术指标：包括通带中心频率 Ω_Z，通带宽度B，通带波动 R_p，阻带衰减 A_s；

（2）按照通带宽度 $\Omega 0_c=1$，通带波动 R_p，阻带衰减 A_s。设计归一化低通滤波器 $G(s0)$；

（3）由(8.4.9)式直接将 $G(s0)$ 转换成带通滤波器 $H(s)$。

8.4.4　低通到带阻的频带变换

低通与带阻滤波器的幅频特性如图8.4.4所示。归一化低通与带阻滤波器的幅度特性如图8.4.5所示。图中 Ω_{c1} 和 Ω_{c2} 分别称带阻滤波器的阻带下边频和上边频。它们是阻带的特征频率。前面已经说过，它可以取 Ω_p，Ω_s 或它们的几何平均值。

带阻滤波器也用阻带中心频率$\pm\Omega_z$和阻带带宽 B 者两个参数来表征。B 通常为作归一化的基准。这两个参数的计算式如下

$$B=\Omega_{C2}-\Omega_{C1} \tag{8.4.10}$$

$$\Omega_z^2=\Omega_{C1}\Omega_{C2} \tag{8.4.11}$$

还是需要寻找一种变换，使得低通滤波器频率轴上 $\Omega 0=\infty$的点变为带阻滤波器 的两个中心频率$\pm\Omega_z$，而 $\Omega 0=\pm 1$的点分别变为带阻滤波器正负两个阻带的上下边频，并保证其带宽为B。$\Omega 0=0$的点应变为带阻滤波器的 $\Omega=\pm\infty$。

低通滤波器的全频带转换为带阻滤波器的正半频率区间。由于带阻滤波器频率特性也必须是对称的，它的负半频率区间也必须由低通滤波器映射而成。因此这一个变换也必须是一对二的多值映射，可以想像它也应该是 Ω的二次函数，在开方时取正取负都能满足。

研究说明，下列二次函数的映射能满足要求：

$$\Omega 0=\frac{B\bullet\Omega}{\Omega^2-\Omega_z^2} \tag{8.4.12 a}$$

其逆函数为

$$\Omega=\frac{-B\pm\sqrt{B^2+4\Omega 0^2\Omega_z^2}}{2\Omega 0} \tag{8.4.12 b}$$

在 B=1 时，按这个函数画出的 $\Omega 0\sim\Omega$曲线如图 8.4.6 所示。它也由上下两段曲线组成，每一段曲线本身是不连续的，在 $\Omega 0=0$ 处有一个从 $0\sim\infty$（或从$-\infty\sim 0$）的突跳。这两段曲线在 $\Omega 0\to\pm\infty$时分别趋向于$\pm\Omega_z$。图 8.4.6 显示了这个函数如何把低通频带变换为带阻频率。

以处于 $\Omega>0$ 区域内的上一段曲线为例，当 $\Omega 0$ 从$-\infty\to 0\to\infty$时，Ω先从 $\Omega_z\to 0$，再由 0 跳到∞，然后从$\infty\to\Omega_z$。$-1<\Omega 0<1$ 的低频通带对应的是带阻滤波器中靠近 0 和∞的两段频带，而低频阻带对应的却是中心频率 Ω_z左右宽度为 B 的频带。这样就形成了图 8.4.5 所示的正频率区间的带阻特性。对处于 $\Omega<0$ 区域内的下一段曲线也可作同样的分析。因此(8.4.12)式在正负半轴上各映射出一个阻带，其中心频率为$\pm\Omega_z$,带宽为 B。将(8.4.12)式转为复频域，用 $s0=j\Omega 0$ 及 $s=j\Omega$代入，有

$$s0 = \frac{B \cdot s}{s^2 + \Omega_z^2} \tag{8.4.13}$$

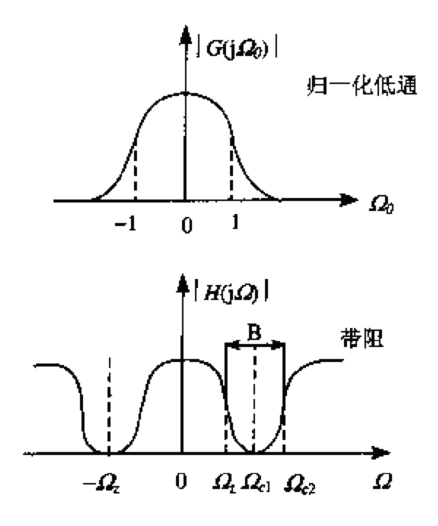


图 8.4.5　归一化低通和带阻滤波器的频率特性对应关系

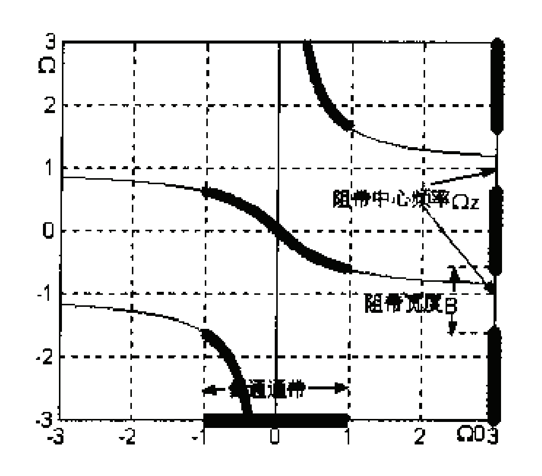


图 8.4.6　带阻和归一化低通的频率映射关系

(8.4.13)式是低通到带阻的频带变换公式。利用该式将带阻的边缘频率转换成低通的边缘频率。因此两者的传递函数之间满足如下关系

$$H(s) = G(s0)\Big|_{s0=\frac{Bs}{s^2+\Omega z^2}} \tag{8.4.14}$$

(8.4.14)式就是由归一化低通直接转换成带阻滤波器的计算公式。

因此，模拟带阻滤波器的设计步骤如下：

（1）确定模拟带阻滤波器的技术指标，包括阻带中心频率 Ω_z，阻带宽度B，通带波动 R_p，阻带衰减 A_s；

（2）按照通带边缘频率 $\Omega 0_c$=1，给定的 R_p 和 A_s，设计归一化低通滤波器 $G(s0)$；

（3）由(8.4.14)式直接将 $G(s0)$ 转换成带通滤波器 $H(s)$。

8.4.5　信号处理工具箱中的频带变换函数

上述的这些变换，都是将变量 $s0$ 代换为 s 的多项式有理分式。所以其编程的原理和双线性变换相仿，读者可以模仿编出相应的程序。本节将介绍信号处理工具箱中的现成函数。

MATLAB 信号处理工具箱提供了从归一化低通滤波器到低通、高通、带通和带阻滤波器的变换函数。只要做到了上节中各滤波器设计的第(1)、第(2)两步，已经设计好了的归一化模拟低通滤波器，知道了它们的分子分母系数向量 num,den，第(3)步就可以直接调用这些函数来求出变换后滤波器的分子分母系数向量 numT,denT。这些函数有四个：lp2lp,lp2hp,lp2bp 和 lp2bs，分别对应于模拟域的低通到低通、低通到高通、低通到带通和低通到带阻四种频带变换。调用格式为：

```
[numT,denT] = lp2lp(num,den,OmegaC)        % 低通到低通频带变换
[numT,denT] = lp2hp(num,den,OmegaC)        % 低通到高通频带变换
[numT,denT] = lp2bp(num,den,OmegaZ,B)      % 低通到带通频带变换
[numT,denT] = lp2bs(num,den,OmegaZ,B)      % 低通到带阻频带变换
```

其中 OmegaC 为待设计滤波器的边缘频率，对低通和高通滤波器通常取为通带边缘频率，也可以取阻带边缘频率或通带和阻带边缘频率的平均值，根据工程问题的具体要求而定。对带通和带阻滤波器，OmegaZ 取为中心频率，B 为带宽。全部的计算过程都由 MATLAB 完成，大大减轻了工作量。但读者应该了解本节所说的原理，知道子程序的来龙去脉，避免囫囵吞枣。

例8.4.1：设根据通带波动R_p=3 dB和阻带衰减A_s=15 dB的要求，已设计好归一化低通滤波器为三阶巴特沃斯型，其传递函数为

$$H_a(s0)=\frac{1}{s0^3+2s0^2+2s0+1}$$

试设计：(a) 通带为 10 Hz 的低通滤波器；(b) 通带下边频为 10 Hz 的高通滤波器；(c) 中心频率为 10 Hz，带宽为 2 Hz 的带通滤波器；(d) 中心频率为 10 Hz，阻带带宽为 2 Hz 的带阻滤波器。又设采样周期为 0.01 s，求相应的数字滤波器。

解：MATLAB程序hc841的核心语句为：

```
num=1;den=[1,2,2,1];
OmegaC=10*2*pi; B=2*2*pi;
[numT1,denT1] = lp2lp(num,den,OmegaC)        %    (1)变换为模拟低通滤波器
[numT2,denT2] = lp2hp(num,den,OmegaC)        %    (2)变换为模拟高通滤波器
[numT3,denT3] = lp2bp(num,den,OmegaC,B)      %    (3)变换为模拟带通滤波器
[numT4,denT4] = lp2bs(num,den,OmegaC,B)      %    (4)变换为模拟带阻滤波器
T=0.01;
[bd1,ad1]=bilinear(numT1,denT1,1/T)          %    (1)双线性变换为数字低通滤波器
[bd2,ad2]=bilinear(numT2,denT2,1/T)          %    (2)双线性变换为数字高通滤波器
[bd3,ad3]=bilinear(numT3,denT3,1/T)          %    (3)双线性变换为数字带通滤波器
[bd4,ad4]=bilinear(numT4,denT4,1/T)          %    (4)双线性变换为数字带阻滤波器
```

关于模拟频带变换的程序段运行的结果如下（对系数向量中每个元素要分别显示）：

```
numT1 = 2.4805e+005
denT1 = 1   125.66   7.8957e+003  2.4805e+005
numT2 =   1.0000   -0.0000    0.0000    0.0000
denT2 = 1   125.66   7.8957e+003  2.4805e+005
numT3 = 1.9844e+003
denT3 = 1  25.13  12159  2.00420  4.8003e+007  3.9171e+008  6.1529e+010
numT4 = 1  0  11844  0  4.6756e+007  2.3468e-007  6.1529e+010
denT4 =1  25.13  12159  2.0042e+005  4.8003e+007  3.9171e+008  6.1529e+010
```

因此四个滤波器的传递函数依次为：

$$H1_a(s)=\frac{2.4805\times10^5}{s^3+125.7s^2+7.896\times10^3s+2.4805\times10^5}$$

$$H2_a(s)=\frac{s^3}{s^3+125.7s^2+7.896\times10^3s+2.4805\times10^5}$$

$$H3_a(s)=\frac{1.9844\times10^3}{s^6+25.1s0^5+12159s^4+200420s^3+4.8\times10^7s^2+3.92\times10^8s+6.15\times10^{10}}$$

$$H4_a(s)=\frac{s^6+11844s^4+4.676\times10^7s^2+6.15\times10^{10}}{s^6+25.1s0^5+12159s^4+200420s^3+4.8\times10^7s^2+3.92\times10^8s+6.15\times10^{10}}$$

为了检验结果的正确性，最好画出这四种情况的幅频特性，因此在程序hc841中可插入以下语句：

```
[Ha1,Omega1]=freqs(numT1,denT1);
[Ha2,Omega2]=freqs(numT2,denT2);
[Ha3,Omega3]=freqs(numT3,denT3);
[Ha4,Omega4]=freqs(numT4,denT4);
subplot(2,2,1),plot(Omega1,abs(Ha1)),axis([0,100,0,1.1])
subplot(2,2,2),plot(Omega2,abs(Ha2)),axis([0,100,0,1.1])
subplot(2,2,3),plot(Omega3,abs(Ha3)),axis([0,100,0,1.1])
subplot(2,2,4),plot(Omega4,abs(Ha4)),axis([0,100,0,1.1])
```

得出的曲线如图8.4.7所示。

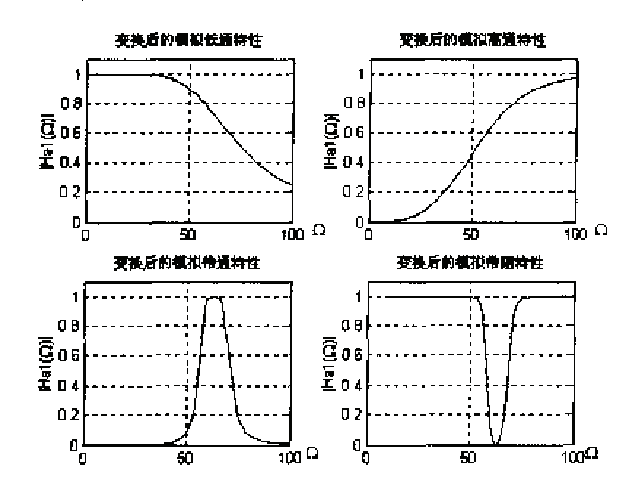


图 8.4.7　用频率变换设计的低通、高通、带通和带阻模拟滤波器的幅特性

核心语句的后四句是对以上设计出的模拟滤波器进行双线性变换，目的是得到其对应的数字滤波器。运行结果为：

```
bd1 =    0.0167    0.0501    0.0501    0.0167
ad1 =    1.0000   -1.7977    1.2211   -0.2898
bd2 =    0.5386   -1.6158    1.6158   -0.5386
ad2 =    1.0000   -1.7977    1.2211   -0.2898
bd3 =    1.0e-003 *
         0.1668   -0.0000   -0.5005   -0.0000    0.5005   -0.0000   -0.1668
ad3 =    1.0000   -4.7347   10.2494  -12.7009    9.4971   -4.0652    0.7956
bd4 =    0.8920   -4.3904    9.8791  -12.7201    9.8791   -4.3904    0.8920
ad4 =    1.0000   -4.7347   10.2494  -12.7009    9.4971   -4.0652    0.7956
```

$$H1(z)=\frac{0.0167+0.0501z^{-1}+0.0501z^{-2}+0.0167z^{-3}}{1-1.7977z^{-1}+1.2211z^{-2}-0.2898z^{-3}}$$

$$H2(z)=\frac{0.5386-1.6158z^{-1}+1.6158z^{-2}-0.5386z^{-3}}{1-1.7977z^{-1}+1.2211z^{-2}-0.2898z^{-3}}$$

$$H3(z)=0.001\cdot\frac{0.1668-0.5005z^{-2}+0.5005z^{-4}-0.1668z^{-6}}{1-4.7347z^{-1}+10.25z^{-2}-12.7z^{-3}+9.497z^{-4}+4.065z^{-5}+0.7956^{-6}}$$

$$H4(z)=\frac{0.892-4.39z^{-1}+9.879z^{-2}+12.72z^{-3}-9.879z^{-4}-4.39z^{-5}-0.892z^{-6}}{1-4.7347z^{-1}+10.25z^{-2}-12.7z^{-3}+9.497z^{-4}+4.065z^{-5}+0.7956^{-6}}$$

为了检验所设计的数字滤波器是否正确，最好计算并画出它们的幅频特性，以便进行直观的判断。可在程序中再增加以下语句：

```
[H1,wd1]=freqz(bd1,ad1);
[H2,wd2]=freqz(bd2,ad2);
[H3,wd3]=freqz(bd3,ad3);
```

```
[H4,wd4]=freqz(bd4,ad4);
subplot(2,2,1),plot(wd1,abs(H1)),axis([0,5,0,1.1])
subplot(2,2,2),plot(wd2,abs(H2)),axis([0,5,0,1.1])
subplot(2,2,3),plot(wd3,abs(H3)),axis([0,5,0,1.1])
subplot(2,2,4),plot(wd4,abs(H4)),axis([0,5,0,1.1])
```

执行这些语句所得出的幅频特性曲线见图 8.4.8。

前面说过，理论上采样周期 T 的大小对于双线性变换的结果并无什么影响。但对计算精度是有影响的，本例的数据可以说明这点。在本书的程序集（所配光盘）中，给出了hc841a和hc841b两个程序，分别把 T 减小和加大十倍。读者运行后可以看出，如果T选得太大，则数字滤波器的边缘频率将向高频端集中；如果 T 选的太小，滤波器的边缘频率将向低频端集中，都会使系数计算误差的影响增大。通常，T 的选择应该使数字边缘频率处在0～π之间的适中位置，比如在$\omega=1$附近，图8.4.8所示的边缘频率参数是较为恰当的。

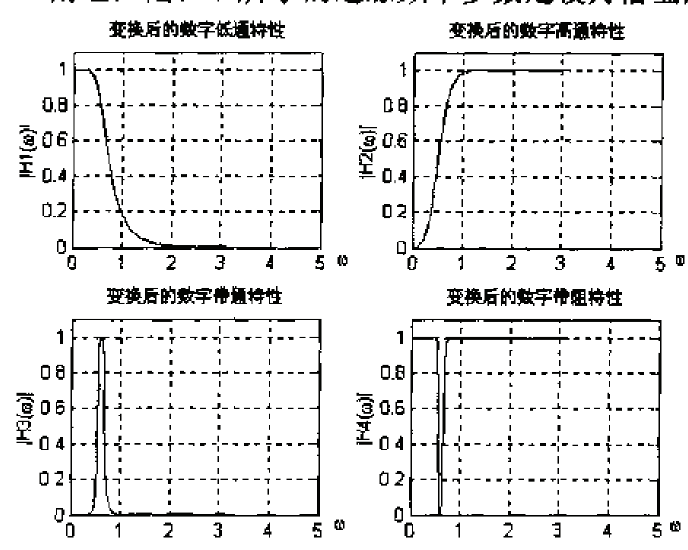


图 8.4.8　经双线性变换后的低通、高通、带通和带阻数字滤波器的幅特性

8.5　数字滤波器的频带变换

根据前几节的讨论，在第 8.2 节中解决了模拟低通滤波器原型的设计问题，在第 8.4 节中研究了把模拟低通滤波器原型变换为模拟低通、高通、带通和带阻滤波器的方法，在第 8.3 节中又解决了把模拟滤波器变换为等价的数字滤波器的方法。因此我们已经可以按图 8.1.3 的方法来进行任何类型的模拟滤波器和数字滤波器的设计问题。MATLAB 的信号处理工具箱为这个设计过程提供了完整的函数体系。

例如，在模拟低通滤波器原型设计方面提供了 butter 和 buttord 等函数；在频带变换阶段，提供了 lp2bp 和 lp2bs 等函数；在模拟数字变换方面，提供了 impinvar 和 bilinear 等函数。书后附录中的信号处理工具箱函数表中，分类列出了所有这些函数的名称。按照所需滤波器的设计要求，有步骤地依次调用这些函数，就可以完成任何幅特性分段为常数的 IIR 选频滤波器的设计任务。

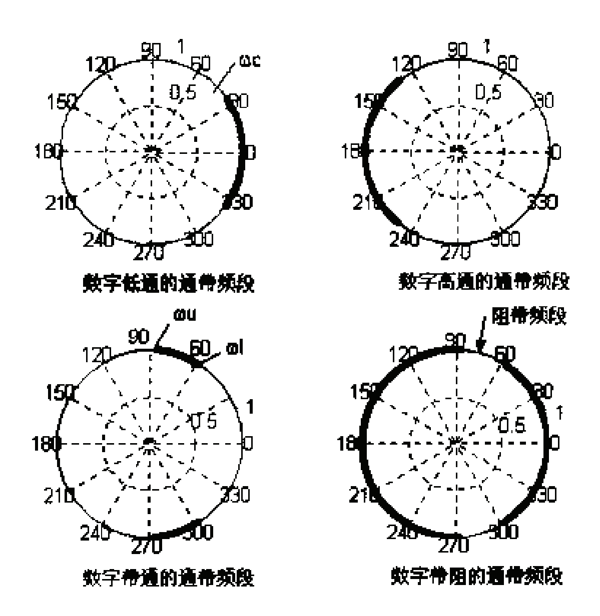
在图 8.1.3 中，还给出了方法 2 的流程。在模拟低通滤波器设计完成之后，它把后两个设计步骤颠倒了一下。即先进行模拟到数字的变换，把模拟低通变为数字低通；然后在数字域进行低通到低通、低通到高通、低通到带通、低通到带阻的频带变换。

方法 2 的设计流程在实践中也有用处。例如已经设计出一个数字滤波器，想把它的频带指标做一些修改调整。如果用方法 1，那就必须先把新的数字滤波器指标变回到模拟域，再在模拟域重新进行频带转换，

然后又做模拟数字变换，才得到修改的滤波器，这就需要三个步骤。如果能够直接在数字域进行频带的修改，那就可省去向模拟域变换和再从模拟域变换回来的两套步骤。

如果按方法 2 的流程进行设计，第一步和方法 1 是一样的；第二步，把模拟低通变换为数字低通，仍然可以调用 impinvar 或 bilinear 函数来完成，也不需要新的函数和方法。关键是在第三步，如何在数字域直接进行频带转换的问题还没有解决。本节就讨论这个问题。

数字频带转换和模拟频带转换相似。差别在于模拟频带 Ω 是在 $-\infty\sim\infty$的虚轴上定义，而数字频带则是在 $-\pi\sim\pi$ 的单位圆上定义的。低通、高通、带通和带阻滤波器的通频带在单位圆上的分布如图 8.5.1 所示。如果能找到适当的从单位圆周到单位圆周的数学变换，把低通滤波器的通带频段变换成其他滤波器所要求的通带频段，那就等于完成了这种滤波器的设计。



注：粗线表示通带频段，在这些频段上的幅特性约为 1。

图 8.5.1　在 z 平面的单位圆上看几种滤波器的通带频率范围

设 $H_{LP}(Z)$为给定的低通数字滤波器原型，又设 $H(z)$是期望的选频数字滤波器，为了避免混淆，相应于 H_{LP} 和 H，选用了大小写不同的复自变量 Z 和 z，分别表示两个平面。在 Z 和 z 平面上的单位圆分别定义为

$$Z = e^{j\omega_{LP}}, \quad z = e^{j\omega} \tag{8.5.1}$$

设从 Z 域到 z 域的映射形式为：

$$Z^{-1} = G(z^{-1}) \tag{8.5.2}$$

则低通滤波器原型 $H_{LP}(Z)$经过这一映射所得到的新滤波器传递函数为：

$$H(z) = H_{LP}(Z)\Big|_{Z^{-1}=G(z^{-1})} \tag{8.5.3}$$

现在来研究对函数 $G(z^{-1})$的要求。假定 $H_{LP}(Z)$是稳定的和因果的滤波器，当然我们也希望 $H(z)$是稳定和因果的。因此，$G(\cdot)$应满足下面的要求：

（1）必须是 z^{-1} 的有理函数，这样 $H(z)$才是可实现的；

（2）Z 平面的单位圆必须映射到 z 平面的单位圆上；

（3）为了保证滤波器的稳定性，Z 平面的单位圆内部必须映射到 z 平面的单位圆内部。

设 ω_{LP}和 ω 分别是 Z 和 z 的频率变量，则要求 2 意味着在各自的单位圆 $Z = e^{j\omega_{LP}}$ 和 $z = e^{j\omega}$ 上有

$$\left|Z^{-1}\right| = \left|G(z^{-1})\right| = \left|G(e^{-j\omega})\right| = 1 \tag{8.5.4}$$

这个函数的幅特性对所有频率都等于 1，这意味着它具有全通型滤波器的特性。5.7 节中讨论过它的一

般形式为：

$$Z^{-1}=G(z^{-1})=\prod_{k=1}^{n}\frac{z^{-1}-\alpha_k}{1-\alpha_k z^{-1}} \tag{8.5.5}$$

而两者的相角之间的关系即表示两个数字频率的映射关系，它也是我们关心的主要指标。因为

$$e^{-j\omega_{LP}}=\left|G(e^{-j\omega})\right|e^{\angle G(j\omega)} \tag{8.5.6}$$

或者 $-\omega_{LP}=\angle G(e^{-j\omega})$

为了满足第 3 个要求，应有$|\alpha_k|<1$。

实际上对于低通和高通，只需取一阶全通函数；对于带通和带阻，也只需取 n=2；适当地选择系数 α_k，就可以满足映射对边缘频率的要求。表 8.5.1 中给出了所需的四种映射以及按边缘频率确定其 α_k 值的公式。

表 8.5.1　数字滤波器的频带变换

变换类型	变　换	参　数
低通[①]	$Z^{-1}\to\frac{z^{-1}-\alpha}{1-\alpha z^{-1}}$	ω_c = 新滤波器的边缘频率 $\alpha=\frac{\sin[(\omega_{cLP}-\omega_c)/2]}{\sin[(\omega_{cLP}+\omega_c)/2]}$
高通	$Z^{-1}\to-\frac{z^{-1}+\alpha}{1+\alpha z^{-1}}$	ω_c = 新滤波器的边缘频率 $\alpha=-\frac{\cos[(\omega_{cLP}+\omega_c)/2]}{\cos[(\omega_{cLP}-\omega_c)/2]}$
带通	$Z^{-1}\to-\frac{z^{-2}-\alpha_1 z^{-1}+\alpha_2}{\alpha_2 z^{-2}-\alpha_1 z^{-1}+1}$	ω_l = 新低边缘频率，ω_u = 新高边缘频率 $\alpha_1=-2\beta K/(K+1)$, $\alpha_2=(K-1)/(K+1)$ $K=\cos[(\omega_u-\omega_l)/2]\cdot\tan(\omega_{cLP}/2)$ $\beta=-\frac{\cos[(\omega_u+\omega_l)/2]}{\cos[(\omega_{uP}-\omega_l)/2]}$
带阻	$Z^{-1}\to\frac{z^{-2}-\alpha_1 z^{-1}+\alpha_2}{\alpha_2 z^{-2}-\alpha_1 z^{-1}+1}$	ω_l = 新低边缘频率，ω_u = 新高边缘频率 $\alpha_1=-2\beta/(K+1)$, $\alpha_2=(K-1)/(K+1)$ $K=\tan[(\omega_u-\omega_l)/2]\cdot\tan(\omega_{cLP}/2)$ $\beta=-\frac{\cos[(\omega_u+\omega_l)/2]}{\cos[(\omega_{uP}-\omega_l)/2]}$

取低通到高通的变换作为一个例子，取 $a=0.3$，分析其 $\omega_{lp}\sim\omega$ 的关系，有

$$e^{-j\omega_{LP}}=-\frac{e^{-j\omega}+0.3}{1+0.3e^{-j\omega}}=\frac{-\cos\omega+0.3+j\sin\omega}{1+0.3\cos\omega-0.3j\sin\omega}$$

进而求得 $-\omega_{lp}=\angle e^{-j\omega_{LP}}=\arctan\frac{\sin\omega}{0.3-\cos\omega}-\arctan\frac{-0.3\sin\omega}{1+0.3\cos\omega}$

给出 ω 的一系列值，即可由此式求出一系列 ω_{LP} 的值。其实用 MATLAB 时根本无需用手工展开上面的函数，直接用如下的三条语句即可：

```
w = linspace(-pi,pi); alpha=0.3;            % 设定频率数组 w
wlp=-angle((j*w+alpha)./(1+alpha*j*w));     % 求映射频率
plot(wlp,w)                                 % 画出逆函数的曲线
```

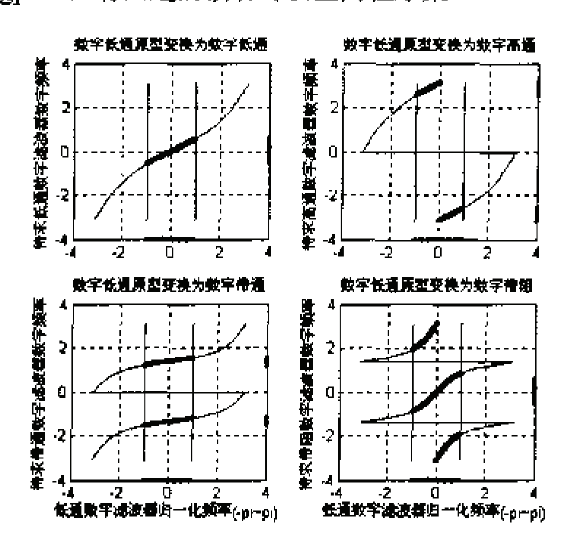
在 $\alpha=0.3$, $\alpha_1=0.3$, $\alpha_2=0.5$ 的条件下，求出四种情况 $Z^{-1}\sim z^{-1}$，即 $\omega_{LP}\sim\omega$ 的数值关系，画成曲线，

---

① 原型低通滤波器的边缘频率为 ω_c。

见图 8.5.2。

在图 8.5.2 的(a)、(b)、(c)、(d)四个子图中，横坐标都是低通原型的数字频率，设它的通带在 -1~1 之间，用粗线表示。把粗线先投影到变换曲线上，成为一段粗的曲线，再把这段曲线横向投影到纵坐标上，表示经频带变换后的滤波器的通带频段。这时可以看出，(a)子图的通带在纵坐标 0 的两边，所以映射后仍为低通；(b)子图的通带变成了纵坐标 π 的两边，所以映射后成为高通；依次可判断(c)子图是带通，而(d)子图是带阻滤波器。此图的绘制程序为 fgp852，有兴趣的读者可以查阅程序集。



注：粗线部分表示低通原型的通带频段，它映射为待求滤波器的通带区

图 8.5.2　四种数字频带变换函数曲线可以生成四种特性的数字滤波器

从上面的例子可以清楚地看出，为了从低通数字滤波器原型得到新数字滤波器的有理函数，应该根据表 8.5.1 实现有理函数代换。这需要进行多项式的多次连乘，若用手工运算是个很麻烦的任务，但是利用 MATLAB 就很方便。因为可以重复地使用 conv 函数来实现连乘。其实在编写双线性变换函数 bilinear0 时我们已经编过这样的程序，那时是用 $\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$ 替换 s，而现在则要由表 8.5.1 中的多项式分式替换 Z。所以程序的编法是相似的。将这个单位圆到单位圆的映射称为 zmapping 函数，它可以用在这四种频带变换中，只要代入不同的 Nz 和 Dz 即可，这个子程序的主要语句如下：

```
function [bz,az] = zmapping(bZ,aZ,Nz,Dz)
-------------------------------------------------------------------------------
% 实现从 Z 域到 z 域的频带变换，将 Z 代换为 Nz(z)/Dz(z)
%  bz,az = 变换后的滤波器分子分母系数向量
%  bZ,aZ = 变换前的滤波器分子分母系数向量
%  Nz,Dz = 变换所用的算子的分子分母系数向量
%
bzord = (length(bZ) -1)*(length(Nz) -1);          % 确定变换后的分子阶数
azord = (length(aZ) -1)*(length(Dz) -1);          % 确定变换后的分母阶数
bz = zeros(1,bzord+1);                            % 分子系数向量初始化为零向量
for k = 0:bzord                                   % 依次求各多项式乘积结果
   pln = [1];
   for l = 0:k-1  pln = conv(pln,Nz);    end      % 求 Nz 的 k 次乘积
   pld = [1];
   for l = 0:bzord-k-1  pld = conv(pld,Dz);  end% 求 Dz 的 bzord-k 次乘积
   bz = bz+bZ(k+1)*conv(pln,pld);                 % 求 bZ*Nz^k*Dz^(bzord-k) 的累加
end
az = zeros(1,azord+1);                            % 分母系数向量初始化为零向量
for k = 0:azord                                   % 依次求各多项式乘积结果
```

```
    pln = [1];
    for l = 0:k-1  pln = conv(pln,Nz);   end       % 求Nz的k 次乘积
    pld = [1];
    for l = 0:azord-k-1  pld = conv(pld,Dz);  end      % 求Dz的bzord-k 次乘积
    az = az+aZ(k+1)*conv(pln,pld);                % 求bZ*Nz^k*Dz^(bzord-k) 的累加
end
az1 = az(1); az = az/az1; bz = bz/az1;            % 将分母多项式首项归一化
```

例 8.5.1：在例 8.4.1 中，已根据通带波动 Rp=3 dB 和阻带衰减 A_s=15 dB 以及 $\Omega_{cLP}=10*2\pi$ [rad / s]的要求，设计了归一化低通滤波器为三阶巴特沃斯型，经双线性变换后，得到的数字低通滤波器传递函数为

$$H1(z)=\frac{0.0167+0.0501z^{-1}+0.0501z^{-2}+0.0167z^{-3}}{1-1.7977z^{-1}+1.2211z^{-2}-0.2898z^{-3}} \tag{8.5.7}$$

现希望用 zmapping 函数实现由低通到高通变换，得到边缘频率为 ω_c=0.3 π 的高通数字滤波器。

解：为了用 zmapping 函数进行由低通到高通变换，要按表 8.5.1 找到 $\alpha=-\frac{\cos[(\omega_{cLP}+\omega_c)/2]}{\cos[(\omega_{cLP}-\omega_c)/2]}$。其中 ω_c 已经给定，而 ω_{cLP} 则需要由原双线性变换的映射关系 $\omega_{cLP}=2\tan^{-1}(\Omega T_{cLP}/2)$ 确定。求出 alpha 以后，得到 Nz = -[alpha,1]; Dz = [1,alpha]; 就可以调用 zmapping 函数。所以本题的 MATLAB 程序 hc851 如下：

```
wc=0.3*pi, OmegaCLP=10*2*pi;
T=0.001;
    wclp=2*atan(OmegaCLP*T/2);                       % 求出原低通数字边缘频率
alpha = -(cos((wclp+wc)/2))/(cos((wclp-wc)/2))  % 低通-高通频带变换参数
Nz = -[alpha,1]; Dz = [1,alpha];                % 低通高通变换分子分母系数向量
bdlp=[0.0167, 0.0501, 0.0501, 0.0167];          % 原低通数字滤波器分子分母系数向量
adlp=[1, -1.7977, 1.2211, -0.2898];
[bhp,ahp] = zmapping(bdlp,adlp,Nz,Dz)           % 用 zmapping 求变换后的数字高通系数向量
```

程序运行的结果为

```
alpha =   -0.7240
bhp =    0.3744   -1.1233    1.1233   -0.3744
ahp =    1.0000   -1.1619    0.6959   -0.1377
```

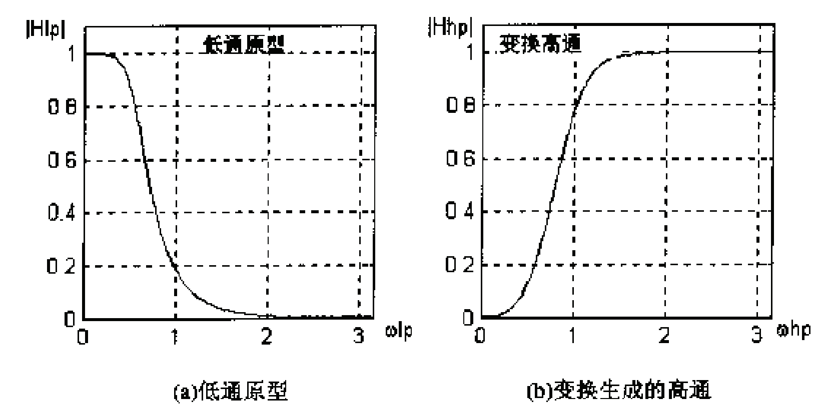
所以所得的高通滤波器的传递函数为：

$$H(z)=\frac{0.3744-1.1233z^{-1}+1.1233z^{-2}-0.3744z^{-3}}{1-1.1619z^{-1}+0.6959z^{-2}-0.1377z^{-3}}$$

再加上以下的主要绘图语句可以帮助检验计算的正确性：

```
[Hlp,wlp]=freqz(bdlp,adlp);            % 求低通滤波器的频率特性
subplot(1,2,1),plot(wlp,abs(Hlp))      %  绘制低通滤波器幅频特性
[Hhp ,w]=freqz(bhp,ahp);               % 求高通滤波器的频率特性
subplot(1,2,2),plot(w,abs(H))?         %  绘制高通滤波器幅频特性
```

执行这些语句得到图 8.5.3 所示的幅特性，可以看出计算的结果是正确的。



(a)低通原型　　　　(b)变换生成的高通

图 8.5.3　低通数字滤波器特性和经数字频带变换生成数字高通滤波器特性

8.6　MATLAB 中直接设计 IIR 数字滤波器的函数

在前几节我们讨论了用来设计数字低通滤波器的四类 IIR 滤波器，即巴特沃斯、切比雪夫 I、切比雪夫 II 和椭圆滤波器设计函数。对每一类，先用求阶次的函数，再用设计函数求模拟低通滤波器，再转换为高通或带通，最后进行模拟—数字变换，共要分四步。从学习它的思路和方法着眼，这是必要的。但从工程应用的角度，还可以简化。MATLAB 将后面三步综合起来，构成 butter,cheby1,cheby2 和 ellip 四个函数，在知道滤波器阶次后，就可以把它们用来直接设计数字高通、带通以及带阻滤波器。这些函数中的频带变换在 s 平面完成，而模拟到数字的转换则用双线性变换。

现在以 butter 函数为例说明其使用方法。根据输入量的不同，它有以下几种形式。

[b,a]= butter (N,wc,'high')：设计 N 阶高通滤波器，wc 为它的 3dB 边缘频率，以 π 为单位，故 0≤w≤1。

[b,a]= butter (N,wc)：当 wc 为具有两个元素的矢量 wc=[w1，w2]时，它设计 2N 阶带通滤波器，3 dB 通带为 w1 ≤w≤ w2，w 的单位为 π。

[b,a]=butter(N,wc,'stop')：若 wc=[w1，w2]，则它设计 2N 阶带阻滤波器，3 dB 通带为 w1≤w≤w2，w 的单位为 π。

如果在这个函数输入变元的最后，加一个变元“s”，表示设计的是模拟滤波器。详细的调用方法，可用 help butter(cheby1,cheby2, ellip)阅读。

为了设计任意的选频巴特沃斯滤波器，必须知道阶数 N 和 3 dB 边缘频率矢量 wc。这可以直接利用信号处理工具箱中的 buttord 函数来计算。如果已知滤波器指标 ω_p, ω_s, R_p, 和 A_s，则调用格式为

[N,wc]=buttord(wp,ws,Rp,As)

对于不同类型的滤波器，参数 wp 和 ws 有一些限制：对于低通滤波器，wp < ws；对于高通滤波器，wp > ws；对于带通滤波器，wp 和 ws 分别为具有两个元素的矢量，wp=[wp1,wp2]和 ws=[ws1,ws2]，并且 ws1 < wp1 < wp2 < ws2；　对于带阻滤波器 wp1 < ws1 < ws2 < wp2。

把 buttord 函数和 butter 函数结合起来，就可以设计任意的巴特沃斯 IIR 滤波器。经过适当修正，对 cheby1、cheby2、ellip 函数可作类似的讨论，读者应当用 help 命令获得详细的说明。对于频率 w，要特别把它的单位弄清楚，如果是模拟频率，单位是 Hz 还是 1/s：如果是数字频率，单位是弧度还是 π 弧度。否则必定出错而且很难发觉。下面将通过一个例子说明这些函数的用法。

例 8.6.1：设计一个切比雪夫 II 型带阻滤波器，它的指标为

$$\omega_{p1}=0.25\pi;\qquad \omega_{p2}=0.8\pi;\qquad R_p=1$$

$$\omega_{s1}=0.4\pi;\qquad \omega_{p2}=0.7\pi;\qquad A_s=40$$

解：根据以上所说的步骤，只要两条核心语句，编成设计程序 hc861 如下：

```
% 数字低通滤波器指标：                  % 切比雪夫-2 型带阻：
ws = [0.4*pi, 0.7*pi];                % 数字阻带边缘频率
```

```
wp = [0.25*pi, 0.8*pi];                  % 数字通带边缘频率
Rp = 1;                                  % 通带波动(dB)
As = 40;                                 % 阻带波动(dB)
% 计算切比雪夫-2 型滤波器参数
[N,wn] = cheb2ord(wp/pi,ws/pi,Rp,As); % 注意频率单位是pi 弧度
[b,a] = cheby2(N,As,ws/pi,'stop');     % 调用数字切比雪夫-2 型带阻滤波器设计函数 cheby2
[sos,G] = tf2sos(b,a)                    % 级联型式实现:
```

程序运行的结果如下。

```
sos = 1.0000   0.3511   1.0000   1.0000   0.2132   0.2145
      1.0000  -0.2434   1.0000   1.0000  -0.4713   0.3916
      1.0000   0.8879   1.0000   1.0000   0.8901   0.4614
      1.0000  -0.5768   1.0000   1.0000  -0.8936   0.7602
      1.0000   1.1456   1.0000   1.0000   1.3041   0.8031
G =   0.1558
```

这是一个十阶滤波器，频域响应图可由以下语句得到：

```
[H,w]=freqz(b,a); plot(w,abs(H))
```

幅频特性曲线见图 8.6.1。

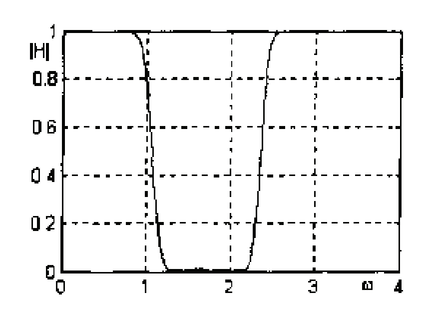


图 8.6.1　例 8.6.1 的切贝雪夫 I 型带阻滤波器的幅特性

8.7　IIR 和 FIR 数字滤波器的比较

前两章讨论了 IIR 和 FIR 两种滤波器传递函数的设计方法。这两种滤波器究竟各自有些什么特点？在实际运用时应该怎样去选择它们呢？下面对这两种滤波器作一简单的比较。

从性能上来说，IIR 滤波器传递函数包括零点和极点两组可调因素，对极点的惟一限制是在单位圆内。因此可用较低的阶数获得高的选择性，所用的存储单元少，计算量小，效率高。但是这个高效率是以相位的非线性为代价的。选择性越好，则相位非线性越严重。FIR 滤波器传递函数的极点固定在原点，是不能动的，它只能靠改变零点位置来改变它的性能。所以要达到高的选择性，必须用较高的阶数；对于同样的滤波器设计指标，FIR 滤波器所要求的阶数可能比 IIR 滤波器高 5~10 倍，结果，成本较高，信号延时也较大；如果按线性相位要求来说，则 IIR 滤波器就必须加全通网络进行相位校正，同样要大大增加滤波器的阶数和复杂性。而 FIR 滤波器却可以得到严格的线性相位。

从结构上看，IIR 滤波器必须采用递归结构来配置极点，并保证极点位置在单位圆内。由于有限字长效应，运算过程中将对系数进行舍入处理，引起极点的偏移。这种情况有时会造成稳定性问题，甚至产生寄生振荡。相反，FIR 滤波器只要采用非递归结构，不论在理论上还是在实际的有限精度运算中都不存在稳定性问题，因此造成的频率特性误差也较小。此

外 FIR 滤波器可以采用快速傅里叶变换算法，在相同阶数的条件下，运算速度可以快得多。

另外，也应看到，IIR 滤波器虽然设计简单，但主要是用于设计具有分段常数特性的滤波器，如低通、高通、带通及带阻等，往往脱离不了模拟滤波器的格局。而 FIR 滤波器则要灵活得多，尤其是他易于适应某些特殊应用，如构成数字微分器或希尔波特变换器等，因而有更大的适应性和广阔的应用领域。

从上面的简单比较可以看到 IIR 与 FIR 滤波器各有所长，所以在实际应用时应该从多方面考虑来加以选择。从使用要求上来看，在对相位要求不敏感的场合，如语言通信等，选用 IIR 较为合适，这样可以充分发挥其经济高效的特点；对于图像信号处理，数据传输等以波形携带信息的系统，则对线性相位要求较高。如果有条件，采用 FIR 滤波器较好。当然，在实际应用中可能还要考虑更多方面的因素。

8.8　MATLAB 中的滤波器设计工具

由于滤波器设计在工程上常常用到，而它的设计相当繁杂，就提出了开发滤波器设计综合性工具的需求。在 MATLAB 环境下，利用它已有的大量滤波器设计函数，加上日益成熟且方便的界面技术，已经可以把所有的设计方法和过程集成在一起，构成一个滤波器综合设计的工具。在信号处理工具箱中，这个工具的名称为 fdatool（Filter Design and Analysis Tool 的缩写）。尽管在学习阶段，我们并不主张学生使用，但考虑到这个工具把滤波器设计的内容进行了概括，通过它有助读者系统归纳第 7、第 8 两章的知识；又考虑到学生已经到了高年级，很快要进入毕业设计，应该知道如何把理论知识转化为工程工具，滤波器设计工具可以当做一个不错的范例；第三也考虑到本书的读者会有不少是工程人员，提供这类工程工具的知识是很有帮助的。当然这里只能作一简单的介绍，目的是引导读者去进一步找参考资料自学。

在 MATLAB 命令窗中，键入 fdatool，就得到如图 8.8.1 所示的界面。

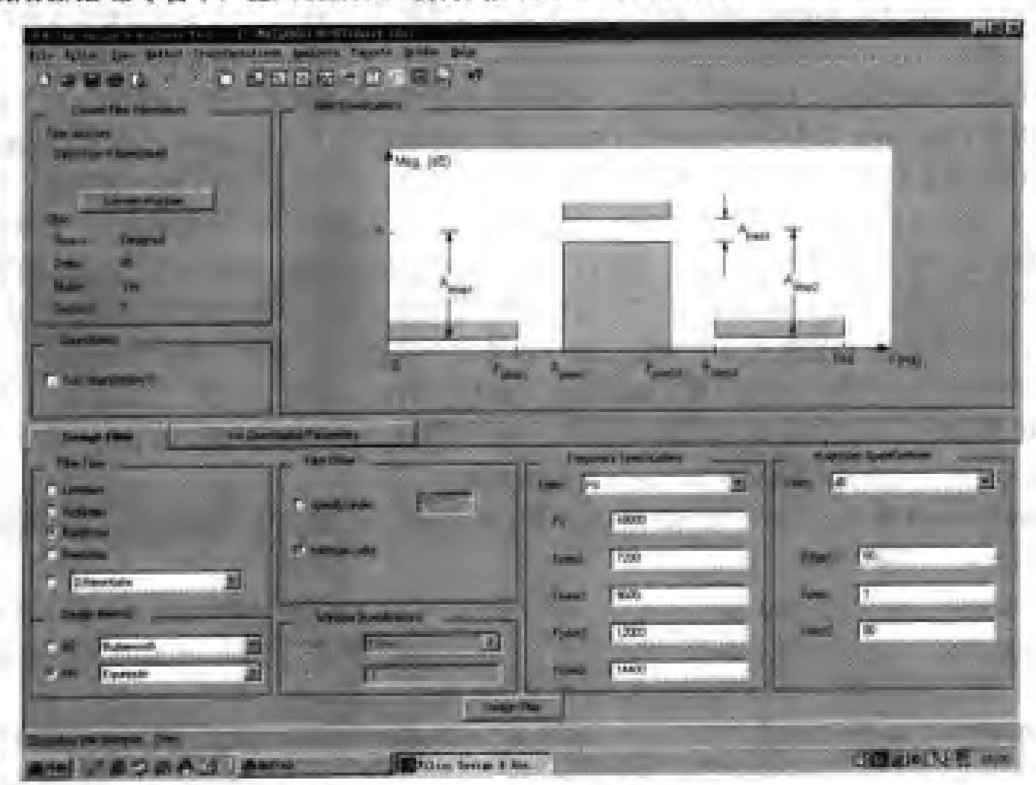


图 8.8.1　滤波器设计工具的启动界面

这个工具的界面包含了全部滤波器设计的功能。大体介绍如下：

界面最上面的一行是文字菜单，第二行是图标按钮菜单；下面是主画面菜单区，它大体也分为上下两半，为了方便，我们分别称之为上、下画面菜单区；最下面正中有一个 【design Filter】按钮，那是在菜单中设定全部参数后，指挥计算机进行设计的确认钮。

文字菜单各项都有下拉式的二级菜单，比如【file】的二级菜单有【new session】,【open session】,【save session】...【print】,【print preview】等，主要用来对本次设计过程命名、建档、存储和输出打印等；其他各菜单项的二级菜单不一一叙述，读者可自行查看和分析。

把文字菜单中常用的二级菜单项突现出来就构成了图标按钮菜单，目的为了更快捷地操作。例如最左边的五个按钮就是【file】项下的五个二级菜单项；再往右的两个按钮是放大和缩小，它们只对图形画面起作用；再往右的十个按钮都是【analysis】项下的二级菜单项，从左向右，第一项为【full view analysis】，它可以把界面中的图形部分取出，单独组成一个视窗；再向右的九个按钮则用来确定图形的内容，依次为

【Filter Specifications】、【Magnitude Response】、【Phase Response】、【Magnitude & Phase Response】、【Group Delay】、【Impulse response】、【Step response】、【Zero/Pole Plot】、【Filter Coefficients】；最右边是【Help】按钮。

上半画面是用来显示设计结果的。其中，右半部是图形画面，它的大小和显示内容受上述按钮的控制。左半部显示当前滤波器的结构。它还设有一个【Convert Structure】按钮，用以使用户能得到不同滤波器结构下的系数。按下这个钮，将出现一个小视窗，其中的复选框提供了八种结构可以选择，确认后，从【Filter Coefficients】钮所得到的就是相应结构下的滤波器系数。这些滤波器系数可以由【file】中的【Export】导出，输出变量的名称可以指定，导出的目标位置可以是 MATLAB 工作空间，或者是其他指定的文件。

另外，它的下部还有一个【Turn Quantization On】(量化)选择框，在开始设计时，它总是空白的，一般是在滤波器初步设计已经完成，要考察系数量化对滤波器性能影响时，再把它打开。这时，图形中将同时显示量化前后滤波器的特性，用户能够方便地看出系数量化对滤波器性能的影响。但是在打开量化钮前应该输入很多参数，如果事先没有设定参数，那么用的是程序中缺省值，不会符合用户的实际。要设定这些参数，先要在下半画面的顶部单击【Set Quantization Parameters】页面标签，这时整个下半画面都会发生改变，后面再作介绍。

下半画面当前处在【Filter Design】页面上，这个页面是用来输入滤波器的设计参数的，它分为四栏：从左到右依次为【Filter type】、【Filter Order】、【Frequency Specifications】、【Magnitude Specifications】。

在【Filter type】栏中，上面五个圆圈是“五选一”的选择框，如果是简单选频类的滤波器，则在前四项中任取一项，如果是其他类，就要选定第五项。然后在它右方的复选框中用下拉菜单来选定具体类型，这里面有数字微分器、希尔波特变换器，还有一些我们未接触过的其他滤波器。再下面的两个圆圈是“两选一”的选择框，用来确定 IIR 还是 FIR 滤波器。它们的右方都有复选框，也用下拉菜单来选定具体类型。IIR 中有巴特沃斯、切比雪夫 I、切比雪夫 II 和椭圆四类；FIR 中则有等波动法、最小二乘法和窗函数法三种，这些都是在本课程中详细介绍的。如果又选了窗函数项，则在它的右方（已进入了第二栏的位置），窗函数选择的复选框将会生效，由灰色变成白色。单击它就可出现下拉菜单，它含有十多项窗函数可供选择，有些窗函数（如 Kaiser 窗）还有可调参数，应填入它下方的参数框中。

在【Filter Order】栏中，只有 “两选一”的两个选择圆圈。一个是由用户强制选择，另一种则是由计算机在设计后自动选择最小阶数。

在【Frequency Specifications】栏中，第一个复选项是选择频率的单位，其中有四个下拉菜单项，分别为 Hz,kHz,MHz 和归一化频率单位。其归一化单位是 π，故奈奎斯特数字频率归一化为 1。以下各框则填入滤波器的边缘频率。框的数目会根据选定的滤波器类别自动变化。

在【Magnitude Specifications】栏中，第一个复选项是选择幅度的单位，其中只有两个下拉菜单项，分别为分贝单位和线性单位。以下各框则填入通带波动和阻带波动的值。对于两频段以上的多频段滤波器，边缘频率指标和对应的幅度指标都应分段标明，因此它们应按 MATLAB 数组格式输入。

实际设计的过程非常简单，先在下半画面的各栏中，正确填写滤波器的要求和指标，然后按最下方正

中的“设计”按钮，结果就会在上半画面中显示出来。需要看何种图形，就可按相应的按钮。取【full view Analysis】时，图形的曲线被放在单独的视窗内，就可以以图形方式存储。需要滤波器系数时，可以如前所述，用【Export】方式，把它导出到适当的地方。

例 8.8.1 现在举一个例子来说明其用法。设滤波器要让(10±2)Hz 的有用调制信号通过，通带的起伏不超过 1dB。要抑制频带小于 6 Hz 和大于 15 Hz 干扰，抑制比要求达到 40 dB 以上。信号的采样频率为 100 Hz，请用 fdatool 工具设计此滤波器，滤波器的类型不限，可比较选择。

解：先选定带通滤波器，在其频率指标项中按 F_s=100 Hz, F_s1=6 Hz, F_{p1}=8 Hz, F_p2=12 Hz, F_s2=15 Hz 填入，幅度栏中填入 Ap=1, As1=As2=40。

在下半界面左边两栏中，先选 FIR 等波动滤波器，指定自动取最小阶数，单击【design】钮，在界面左上方得到的滤波器阶数是 43 阶，同时给出了它的幅特性，可以检验它满足给定的指标要求。

仍在下半界面左边两栏中，选 IIR 切比雪夫 II 滤波器，指定自动取最小阶数，单击【design】钮，得到的滤波器阶数是六阶，同时给出了它的幅特性，可以用放大图形的方法检验它是否满足给定的指标要求。做法是先单击【Zoom In】钮，再在画面上关心的位置附近单击一下，系统就会把该点附近的画面放大。如果选 IIR 巴特沃斯滤波器，得到的滤波器阶数是八阶。

在设计完成以后，可以把量化开关激活，图形框中就会出现两根曲线，其中绿色的就是滤波器量化后的幅频特性。可以发现，在 FIR 设计中，量化前后的幅特性几乎重叠，说明系数按系统的缺省设置进行量化对设计结果几乎没有影响。但对 IIR 设计，两根曲线就“差之千里”了。本应该是通带的区域，量化后滤波器的幅特性却降到了-40dB，而本该是阻带的 35 Hz 附近，幅特性却超过了 0dB。这样的滤波器是根本不能用的。

问题的关键在于滤波器的结构设定，按下【Convert Structure】按钮，可以发现，在默认状态下的滤波器取的是直接 II 型结构。对于六阶的切比雪夫滤波器，分母上是一个六次多项式。从高等代数中知道，高次多项式系数的微小变化会带来根的很大误差；它又处在传递函数的分母上，就是进行递归运算的部位，所以对系统传递函数影响很大。如果把结构改成二阶环节级联形式，重新计算，系数量化前后的幅特性就很相近了，这也说明了系数量化对二阶环节影响较小，所以数字滤波器实现时，特别是它的分母系数，应该采用二阶级联形式。

当需要把量化方式设置为实际系统的状况时，应该单击【Set Quantization Parameters】页面标签，并使【Turn Quantization On】框处于选定状态，这时的下半画面就被激活而成为如图 8.8.2 所示的形式。其中的可变因素有量化内容、量化方法和量化位数三类。量化内容有系数量化、输入量化、输出量化、乘法量化、乘积量化、总和量化等六项，量化方法有四舍五入、向下取整、向上取整和饱和取值等四种，量化位数则可设定 16 位、8 位或其他二进制位数。这方面的问题只给读者提一个头，具体运用必须仔细阅读手册。

在设计完成后，可以单击【file】中的【Export】，此时会弹出一个小的视窗，其中给出待存储变量的缺省名称，用户可以把它修改为自己想取的任何名称。假定我们取量化前的设计结果，第一个是 43 阶 FIR 滤波器的分子，分母系数向量，设其变量名为 Num1 和 Den1；第二个是六阶切比雪夫 II 型 IIR 滤波器，用二个二阶环节级联方式导出，其变量命名为 SOS2 和 G2，单击“OK”钮确认后，这些变量可以都导出到 MATLAB 工作空间中。然后在命令窗中键入 whos，MATLAB 会显示这几个被导入的变量名称。

```
Name      Size      Bytes   Class
Den1      1x1       8   double array
G2        1x1       8   double array
Num1      1x44      352   double array
SOS2      3x6       144   double array
```

键入变量名，就可以得到系数的具体值。

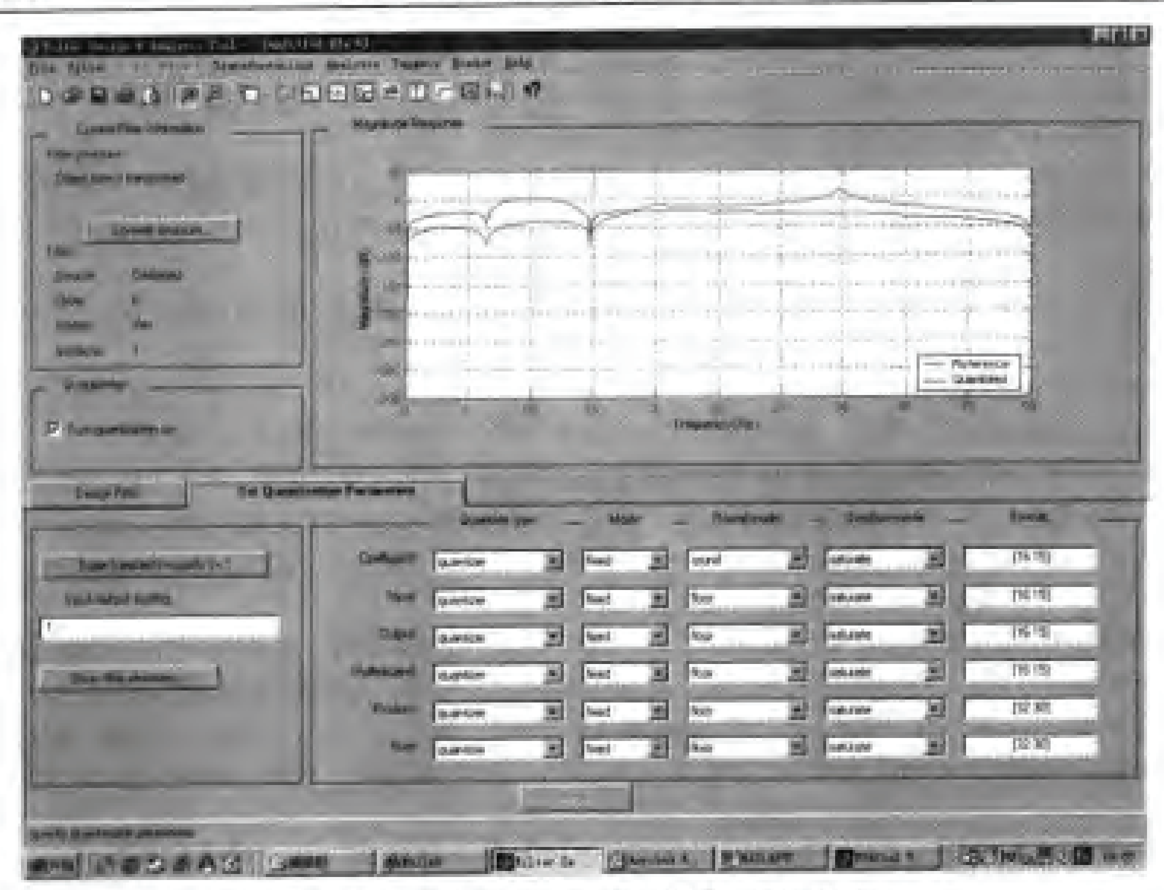


图 8.8.2　滤波器设计工具的量化设置和分析界面

复习思考题

8.1 模拟滤波器的平方幅特性 $J(\Omega)$的指标 ε 及 A 的定义是什么？它们和第 7.1 节中的 FIR 数字滤波器指标有何数学关系？

8.2 在 s 平面上的平方幅函数$|\mathrm{Ha}(s)\mathrm{Ha}(-s)|$的零点和极点分布有什么特征？应该怎样选择其中的零极点来组成一个稳定的最小相位滤波器？

8.3 巴特沃斯滤波器的频率特性有什么特点？它的两个参数 Ω_c 和 N 对频率特性各有什么影响？哪个影响带宽？哪个影响过渡带宽？这两个带宽的定义各是什么？给定带宽和过渡带宽，是否就可以确定巴特沃斯滤波器的两个参数 Ω_c 和 N ？

8.4 切比雪夫滤波器的带宽是否也是这样定义的？知道了它的带宽和比值 Ω_s/Ω_p 和阻带的衰减 Ap，如何确定切比雪夫滤波器的参数 N,ε 及 Ω_c？

8.5 对巴特沃斯、切比雪夫 I、切比雪夫 II 及椭圆滤波器四种情况的幅特性特征（如通带和阻带的波动性，过渡带宽等）进行综合比较，列表说明。

8.6 把模拟滤波器变换为数字滤波器的等价原则是什么？脉冲响应不变法和双线性变换法遵循的原则有何不同？变换前后的两个滤波器的时域脉冲响应 $h_a(t)$ 和 $h(n)$应近似满足什么关系？是 $h(n)=h_a(t)\big|_{t=nT}$ 还是 $h(n)=T\cdot h_a(t)\big|_{t=nT}$ ，为什么？它们的频率特性又应近似满足什么关系？

8.7 能否用脉冲响应不变法由模拟高通滤波器变换为数字高通滤波器？能否用双线性变换从相应的模拟滤波器设计数字高通，带通或带阻滤波器？

8.8 从脉冲响应不变法设计的数字滤波器对照推想，能否得出阶跃响应不变法的思路和

应满足的关系式。

8.9 归纳出由归一化低通做四种频带变换得到四种滤波器的规律（包括模拟域和数字域），列表说明。

8.10 从已知模拟低通原型滤波器变换到任意频带的数字滤波器要经过哪些步骤？各个步骤中所用的理论依据是什么？有哪些工具箱函数或自编函数可以利用？对两种方法分别说明。

8.11 从阶数、时延、线性相位、稳定性、运算量和存储单元等几个方面，对 FIR 和 IIR 滤波器做一比较。

习题

8.1 设计巴特沃斯模拟低通滤波器，满足下列指标：通带边缘频率 $f_p = 6$ kHz，通带最大波动 $R_p = 3$ dB，阻带边缘频率 $f_p = 12$ kHz，阻带最小衰减 $A_s = 25$ dB，求出滤波器的归一化传递函数和非归一化传递函数。

8.2 设计一个切比雪夫 I 型模拟低通滤波器，满足下列指标：通带边缘频率 $f_p = 6$ kHz，通带最大衰减 $R_p = 1$ dB，阻带边缘频率 $f_p = 12$ kHz，通带最大衰减 $A_s = 50$ dB，求出滤波器的归一化传递函数和非归一化传递函数。

8.3 设计一个模拟切比雪夫 II 型低通滤波器，它在 250 Hz 处具有 0.5 dB 或更好的波动，在 300Hz 处至少有 45 dB 的衰减。画出滤波器的幅度响应、对数幅度响应、相位响应和脉冲响应图。

8.4 信号 $x_a(t)$ 包含两个频率：100 Hz 和 130 Hz，希望抑制 130 Hz 的分量至 50 dB，同时以小于 2 dB 的衰减允许 100 Hz 的分量通过。设计一个阶数最小的切比雪夫 I 型模拟滤波器实现此滤波，画出对数幅度响应图以验证此设计。

8.5 直接调用 MATLAB 函数，设计一个四阶巴特沃斯模拟带通滤波器，其通带边缘频率为 $fp1 = 8$ kHz，$fp2 = 11$ kHz，算出它在这两个边缘频率处的幅度，并画出其频率响应进行校核。

8.6 已知模拟滤波器的传递函数为 $H_a(s) = \dfrac{1}{20s^2 + 9s + 1}$，采样频率为 $F_s = 10$ Hz。

(a)　试用脉冲响应不变法，求出与其等价的数字滤波器传递函数 $H(z)$。求出原模拟滤波器和变换后的数字滤波器的脉冲响应，进行比较。

(b)　用双线性变换法，重复(a)中的工作。并对两种方法进行比较。

(c)　对变换前后的滤波器频率响应进行比较。

8.7 将一个四阶巴特沃斯模拟低通滤波器用脉冲响应不变法变换为一个数字低通滤波器，它的采样频率为 100 Hz，在通带边缘频率为 30 ad / s 处的幅特性为 3 dB。画出模拟和数字滤波器的频率响应并作比较。

8.8 同题 8.7，只是改用双线性变换，求其结果并进行比较。

8.9 设计巴特沃斯数字低通滤波器满足下列指标：通带边缘频率：0.4π，$R_p = 0.5$ dB；阻带边缘频率：0.6π，$A_s = 50$ dB。采用脉冲响应不变法和双线性变换法，设 T＝0.01。求出有理函数形式的系统函数，画出对数幅度响应、脉冲响应 $h(n)$ 和模拟原型滤波器的脉冲响应 $h_a(t)$ 图，并比较它们的形状。

8.10 用椭圆原型设计数字低通滤波器，满足下列要求：通带边缘频率：0.4π，R_p=1dB；阻带边缘频率：0.5π，A_s = 60 dB。用 impinvar 函数和 bilinear 函数分别进行设计，并对结果进行比较。

8.11 用切比雪夫 I 型设计一个带通 IIR 数字滤波器，满足下列指标：

$$0.95 \leqslant |H(\omega)| \leqslant 1.05, \quad 0 \leqslant |\omega| \leqslant 0.25\pi$$

$$0 \leqslant |H(\omega)| \leqslant 0.01, \quad 0.35\pi \leqslant |\omega| \leqslant 0.65\pi$$

$$0.95 \leqslant |H(\omega)| \leqslant 1.05, \quad 0.75 \leqslant |\omega| \leqslant \pi$$

利用 cheby1 函数进行设计，求出滤波器的系统函数 $H(z)$，画出对数幅度响应图和脉冲响应图。

8.12 利用双线性变换方法，设计一个带宽为 0.08π 的十阶椭圆阻带滤波器以滤除数字频率为 ω=0.44π 的信号，选择合理的阻带衰减值，画出幅度响应。然后，令序列

$$x(n)=\sin[0.44\pi n], \quad n = 0, \cdots, 200$$

的 201 个样本，通过此带阻滤波器，求滤波器的输出。解释所得的结果。

8.13 将题 8.7 的数字低通滤波器，用数字频率域的直接变换法，变为一个数字带通滤波器。该滤波器的 3 dB 通带边缘频率为 100 rad/s 和 180 rad/s，采样频率为 120 Hz。注意变换前后的两个滤波器具有不同的采样频率。

8.14 设计数字低通滤波器，以满足下列要求：通带边缘频率：0.3π，R_p = 0.5dB；阻带边缘频率：0.4π，A_s = 50 dB。

(a) 利用 butter 函数进行设计，求出阶数 N 和实际的最小阻带衰减的 dB 数；

(b) 利用 cheby1 函数进行设计，求出阶数 N 和实际的最小阻带衰减的 dB 数；

(c) 利用 cheby2 函数进行设计，求出阶数 N 和实际的最小阻带衰减的 dB 数；

(d) 利用 ellip 函数进行设计，求出阶数 N 和实际的最小阻带衰减的 dB 数；

(e) 比较上述各种设计的阶数，实际最小阻带衰减及群延迟。

第9章　综合应用

本书在第2～4章中，介绍了信号的时域和频域的分析和计算；在第5～8章中又介绍了系统（滤波器）的分析和设计；这些都是从事信号处理的必要基础，不过所涉及的范围还不宽，例如还只限于一维信号、还没有涉及随机过程、还没有谈及非线性信号处理等等，而信号处理技术的发展实在太迅速了，比如数字电视已经在2003年开始进入中国人的生活，移动通信制式和手机类型的更新使人目不暇接，这些巨大的技术更新都与数字信号处理理论和技术的发展紧密相关，更难预计五年或十年以后还要提出哪些要求。

本书前8章是按目前大学本科教学的基本要求编写的，有些知识应该在研究生的阶段再进一步学习，不宜都压到本科教学中来。我们认为，在本科学习阶段，在掌握了一定的基础知识之后，比较重要的问题是学会联系实际，灵活应用。信号处理的概念比较深，其中许多概念不是单纯用数学推导就能掌握的，往往要通过在实践中反复运用，才能真正发展和掌握。所以我们开辟这一章，目的就是通过一些实例来说明如何把理论用到工程实践中去。同时又加深了对理论的理解。另外在本章中还要介绍一些MATLAB的工具，使读者在遇到新问题的时候，知道从什么方面还能找到适当的工具，帮助自己解决问题。

本章先介绍信号处理工具sptool，它是MATLAB信号处理工具箱中的一个集成环境。其中综合应用了工具箱中的函数，使用户能最方便和快速地解决信号时域、频域的分析问题以及信号通过滤波器的效果分析。学习和使用这个工具可以帮助读者把已学的信号处理知识系统化和集成化，而且对于在工程中进行信号处理会有很大的帮助。我们建议读者都看看这一部分，帮助自己对本课程的内容做一归纳。

随后我们将介绍几个信号处理的应用实例：第9.2节介绍信号的量化问题，信号的量化通常用A/D变换，这是一种绝对（定点）量化的方法。其特点是小信号时相对误差加大，本节将介绍用压缩扩张器减小相对误差的非线性处理方法；第9.3节介绍滤波器系数量化和计算量化的影响，系数量化通常用相对（浮点）量化，本节给出了数组设置和计算的相对量化的MATLAB程序，为方便地进行数学仿真创造了条件；第9.4节介绍双频拨号系统中的数字信号处理，它是一个从双频发送到接收检测的完整的系统，其中包括DFT的goertzel算法开发；第9.5节介绍了频谱分析仪中的数字信号处理技术，实际上主要涉及分辨率等参数的选择，同时提出了关于非平稳信号谱分析的概念；第9.6节介绍音响系统中的信号处理，它涉及回声和混响的生成和处理；第9.7节介绍变采样率系统，包括内插和抽取，这也属于一种非线性处理，会产生新的频谱分量，所以也是对本书中原有概念的扩展；第9.8节研究一个简单的空域信号处理问题，使读者把时域信号处理扩展到空域信号处理。

上述的七节都是以前面学过的八章为基础，但都把原有的知识拓宽和加深了一步。在本书中，对每一个问题都只作了简要的提示，各个问题都有很大的展开余地，所以其中任何一个问题都可以作为实验或课程设计的基础。这要由教师和读者自行取舍，进行适当的开发。第9.9节是结束语，大概地介绍了一下数字信号处理的应用领域，并介绍了MATLAB中与信号处理有关的各种工具箱函数。

9.1　信号处理工具的介绍

MATLAB 信号处理工具箱还为信号处理的研究工作提供了一个集成环境和工具。这个信号处理工具的名称为 sptool（就是 Signal Processing Tool 的缩写）。大家知道，信号处理主要有两大任务：一是对信号进行分析，就是在时域分析它的波形和在频域分析它的频谱；二是滤波器设计，也就是根据对输入输出信号的要求，设计实现这个要求的系统。在这两个任务进行过程中，经常要做第三个任务，那就是进行检验，要把信号加到所设计的滤波器中，看它的输出是否满足要求。Sptool 把这三个任务集成在一起，加上了适当的管理功能，配以良好的工作界面，利用它可以大大提高研究工作的效率。考虑到本书的读者会有不少是工程人员，所以提供这类工程工具的知识是很有帮助的，当然这里只能作一简单的介绍，目的是引导读者去进一步找参考资料自学。

在 MATLAB 命令窗中，键入 fdatool，就得到如图 9.1.1 所示的界面。其中分为【Signals】、【Filters】和【Spectra】三个栏目，每个栏目中分别存入了系统中原来已保存的信号、滤波器和频谱的名称。下面先来看如何把所需的信号和滤波器输入这个环境中去。

图 9.1.1　信号处理工具 sptool 的启动界面

（1）信号和滤波器的导入

先在 MATLAB 工作空间中放入所需的分析的信号和滤波器。设滤波器就是例 8.8.1 中设计并导入了工作空间的滤波器，再来建立一组信号。在命令窗中，键入：

```
n=0:200;T=1;s1=sin(5*2*pi*n*T);s2=sin(10*2*pi*n*T);
s3= cos(15*2*pi*n*T);s=s1+s2+s3;
```

这时 s1,s2,s3 和它们的合成信号 s 都已输入工作空间。

单击左上角的菜单项【File】及其下拉菜单中的【Import】，出现新的视窗如图 9.1.2 所示。它分成三个栏目，左边一栏为【Source】，其中有【Workspace】或【File】二选一圆圈，选定【Workspace】时，中间栏的【Workspace Contents】将显示工作空间中的全部变量，如图 9.1.2 所示；如果选择【File】作为数据来源，则下方的框中应键入文件名及其路径。右边的栏指定导入的目标，即要说明导入的数据是作为信号、滤波器还是频谱。在图示状态，顶部的下拉选择框中显示的是【Signal】，如果我们要把合成信号导入，就在直接的工作空间内容栏中选定 s，单击[→]，它就进入右边的数据框中。右栏的下部有【Sampling Frequency】和【Name】两个框需要用户填写。其中【Name】栏已写上默认的变量名，它是在 sig 后面加上新的序号，当前它是 sig7，用户也可以另起名称改写。

要输入滤波器参数时，首先在右栏的上方下拉菜单选择框内改选【Filter】，此时它的下方将出现【Form】框，它的下拉菜单中，有【Transfer Function】(传递函数)、【2nd Ordr Section】(二阶级联)等四种可选项。选【Transfer Function】时，下面就出现【Numerator】(分子)和【Denominator】(分母)两个框，可以用前面说过的方法分别把工作空间变量 Num1 和 Den1 填入框中，然后在右下方的两个框中分别键入采样频率和滤波器名称后，单击【OK】，就完成了滤波器的导入。默认名称是在 filt 后面加上新的序号，用户也可以另起名称改写。选【2nd

Ordr Section】时，下面就出现【Sos】框，将工作空间变量 sos2 填入，照样办理即可。这样在例 8.8.1 中设计的 FIR 和 IIR 滤波器分别成为信号处理工具中的 filt7 和 filt8。

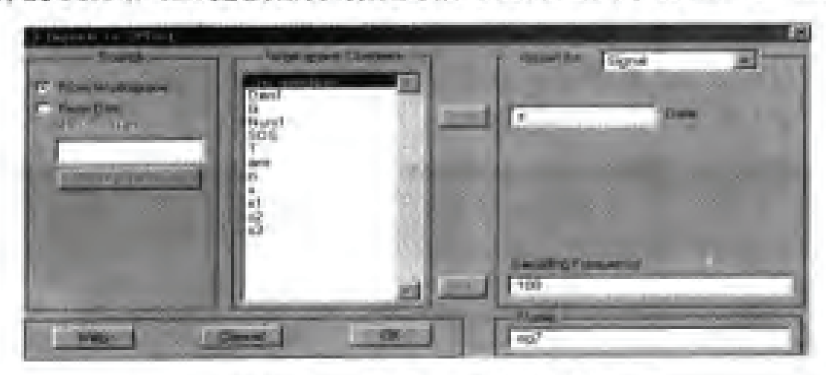


图 9.1.2 sptool 的数据导入界面

（2）信号的时域和频域观测

在图 9.1.1 上，选定 sig7，单击【view】(观测)，进入【Signal Browser】(信号浏览器)视窗，如图 9.1.3 所示。

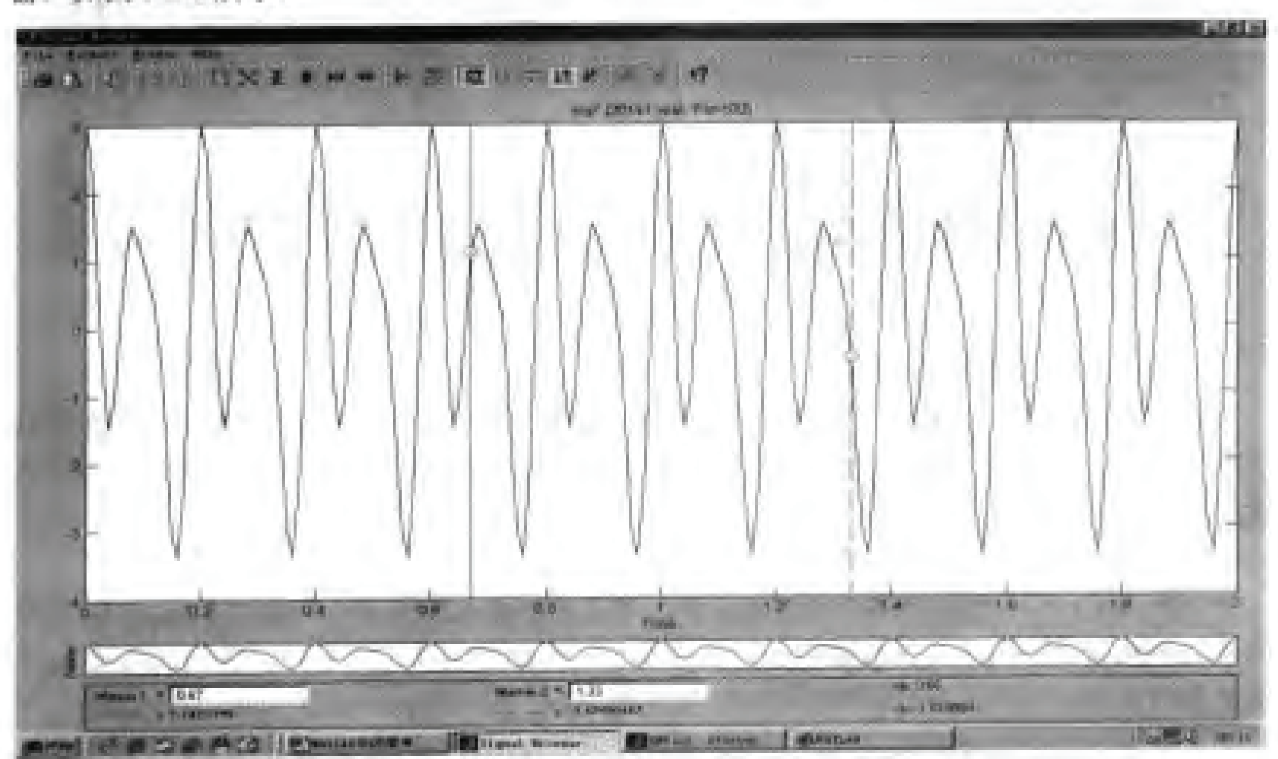


图 9.1.3 信号浏览器（Signal Browser）的界面

在这个界面上有许多帮助测量和输出信号特性的菜单和按钮。主菜单上【File】下主要是打印命令，【Markers】用来在图上设置标志，下有【Vertical】(垂直标线)、【Horizontal】(水平标线)、【Track】(跟踪点)和【Slope】(斜率线)，还有峰点和谷点显示。主菜单的下面一行图标按钮，从左到右的大体功能依次为，打印和预览、把信号变为声音、x 和 y 方向的放大和缩小、标志点和线的设定和去除等。把鼠标移到按钮上，它会出现英文名词，因此这里无需赘言。界面的最下方则给出标志的读数。在它的上方，是一个高度很小的图形窗，它给出信号的全景，在主图形 x 方向放大时，它仍表示全部信号，而且用方框指明主图形信号在全信号中的位置。

要想求得该信号的频谱，也先选定 sig7，然后在频谱栏下单击【creat】,此时出现频谱观测器的界面，见图 9.1.4。先要在左栏上部选定求频谱的方法，目前我们只学了一种，即选 FFT。在下面的框中填入点数，例如 1024，然后单击【Apply】，就出现了频谱的曲线。它有三个峰值，恰好就对应了输入的三个正余弦信号分量。

此界面上的菜单与按钮和信号观测器相似，不再重复叙述。

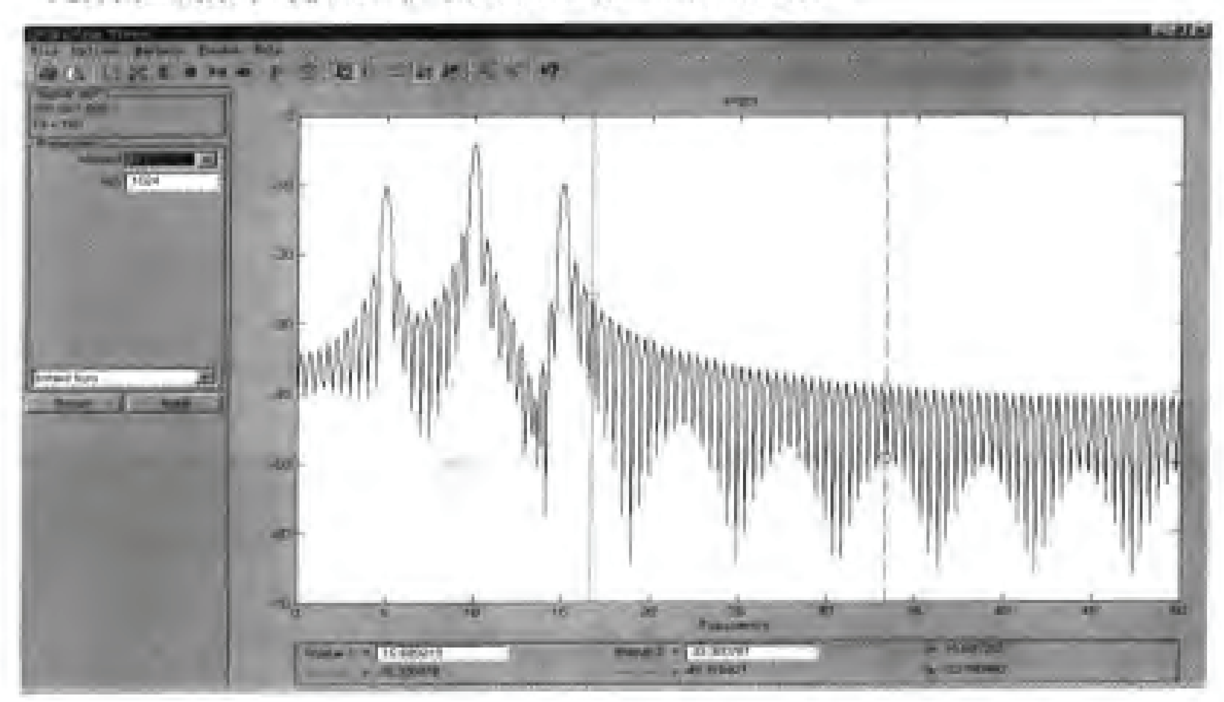


图 9.1.4　频谱观测器（Spectrum Browser）的界面

（3）滤波器的观测和修改

选定滤波器的名称 filt7，单击其下方的【View】，就可以观测它的频率响应。如单击其下方的【Edit】，那就不仅能观测，还能修改。它的功能不如滤波器设计工具那样全，但大体接近，读者不难掌握，这里就从简了。

（4）让信号通过滤波器求输出信号

在图 9.1.1 中，选定信号和滤波器名称，单击滤波器栏的最下一个按钮【Apply】，这意味着将所选信号加到所选滤波器中去。此时将产生一个小视窗，提示用户输入信号为 sig7、滤波器的名称为 filt7，并提供输出信号默认名称 sig8。如无修改，单击 OK，则输出信号将自动生成并列写在信号栏中。对它进行观测就可以知道所设计的滤波器是否完成了给定的各项指标。在本例中 sig8 基本上是一个较纯的 10 Hz 正弦波，这是不难想像的。经过仔细分析，如不满意，可以修改滤波器后重新实验。

用这个工具，不难在很短的时间内完成多个滤波器的设计，并分别观测它们的输出。所有的结果都存储在案，可用于试验报告。所以这是一个提高效率的有效工具。

9.2　语音数字化量化噪声的改善

脉冲编码调制（Pulse Code Modulation，PCM）是把模拟信号量化为二进制数的最简单的方法。以 N 个脉冲表示 N 位二进制数，以脉冲的有无判断它是 0 或 1。它也是用数字方式

传输或存储信号的常用方法之一。PCM 被广泛应用于电话通信中的语音传送和利用无线电传输的遥测系统中。

通过电话线传输的语音信号频带限于 4kHz 以下的频率范围。因此其采样频率取 8kHz(样本数/秒)，并用 N 位二进制序列表示它的值，每个样本量化为 2^N 个电平之一。所以，传输数字化语音信号所要求的速率为每秒 $8000\times N$ 位。

量化处理的数学模型为

$$x_q(n)=x(n)+q(n) \tag{9.2.1}$$

其中 $x_q(n)$表示 $x(n)$的量化值，$q(n)$表示量化误差，将其看做一加型噪声。假设采用的是均匀的量化器，则可用如下均匀概率密度函数 $p(q)$统计描述量化噪声特性:

$$p(q)=\frac{1}{\Delta},\quad -\frac{\Delta}{2}\leqslant q\leqslant\frac{\Delta}{2} \tag{9.2.2}$$

其中，量化器的量化步长为$\Delta=2^{-N}$。不难算出，量化误差的均方值为:

$$E(q^2)=\frac{\Delta^2}{12}=\frac{2^{-2N}}{12} \tag{9.2.3}$$

用分贝来度量的噪声均方值为:

$$10\log E=10\log\left(\frac{2^{-2N}}{12}\right)=-6N-10.8\ \text{dB} \tag{9.2.4}$$

可以看出，上述的量化器每增加一位，量化噪声减小 6 dB，高质量语音要求每个样本至少量化为 12 位，因此要求的传送速率至少为 96000 bit / s（位/秒）。

输入最大幅度为±V 伏的 N 位（不含符号位）二进制 A/D 变换器的数学模型，可以根据第 1 章中的公式（1.1.1）建立。令其中的量化步长 $\Delta x=\dfrac{V}{2^N-1}$，得出如下的二进制量化子程序 bqtize。

```
function y=bqtize(x,N,V)
if nargin<3 V=max(abs(x));end          % V 默认时，取 x 的最大值为 V
ax=abs(x);                             % 去掉符号
deltax=V/(2^N-1);                      % 求量化步长
xint=fix(ax./deltax+0.5);              % 将 x 用 deltax 四舍五入量化为整数
y=sign(x).*xint.*deltax;               % 再将此整数缩小 2^-(N-m)，恢复量化后的原值
```

这个 A/D 变换子程序的输入是连续模拟电压 x，输出则是量化了的模拟电压 y。均匀量化器在信号的整个动态范围中的量化步长相同，所以量化噪声均方值不变。然而，语音信号的特性是小幅度比大幅度出现得频繁。对小信号而言，量化噪声使信噪比大大下降。解决的途径之一是用非均匀量化器，这种量化器对小的信号幅度提供较密的量化步长，对大的信号度提供较粗的量化步长。这样，其量化误差的均方值就可以小于式(9.2.4)给出的值。不过在技术上制造非均匀量化器的芯片是困难和昂贵的。得到非均匀量化器特性的另一个方法是用压缩-扩张器。可先使信号通过压缩幅度的非线性器件，后面再接一均匀量化器，再用逆向扩张幅度的非线性器件恢复信号，如图 9.2.1 所示。例如，在美国和加拿大的通信系统中使用的对数压缩器(称为 μ 律压缩器)具有如下输入输出幅度特性:

$$y=\frac{V\ln(1+\mu|x/V|)}{\ln(1+\mu)}\text{sign}(x)\qquad |x|\leqslant V,|y|\leqslant V \tag{9.2.5}$$

其中，x 是归一化输入，y 是归一化输出，sign(·)是符号函数，μ 是控制压缩特性的参数。

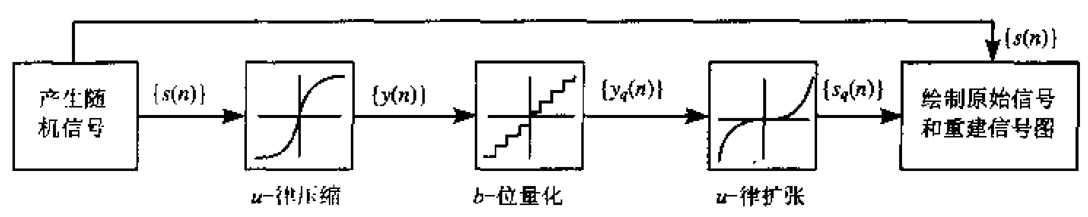


图 9.2.1　压缩器、A/D 变换器和扩张器的连接框图

在欧洲的通信系统中所用的对数压缩标准为 A-律，

$$y=\begin{cases} V\dfrac{1+\ln(A/V\cdot|x|)}{1+\ln(A)}\operatorname{sign}(x) & \dfrac{V}{A}\leqslant|x|\leqslant V \\ \dfrac{A\cdot|x|}{1+\ln(A)}\operatorname{sign}(x) & 0\leqslant|x|\leqslant\dfrac{V}{A} \end{cases} \tag{9.2.6}$$

其中 A 选为 87.56，(9.2.5)和 (9.2.6)两种压缩特性非常相似。

在数字信号的接收端，由脉冲编码调制(PCM)数字信号量化值恢复模拟信号时，要求解码器用逆对数关系来扩展信号幅度，例如，在 μ -律中，逆关系如下：

$$x=\frac{V(1+\mu)^{|y/V|}-1}{\mu}sign(y) \tag{9.2.7}$$

组合成对的压缩-扩张器简称为压扩器。

我国采用的是 μ-压扩器编码。其中，选定 $\mu=255$ 为标准值。与均匀量化器相比，该 μ 值的效果是使量化噪声功率降低 24 dB。所以，一个与 $\mu=255$ 的对数压扩器连用的 8 位量化器产生的语音质量与无压扩器的 12 位量化器相同。因此，压扩的 8 位 PCM 语音信号具有 64000 bit / s 的速率。

把式(9.2.5)和式(9.2.7)分别写成压缩器函数程序 mulawcom 和扩张器函数程序 mulawexp，其调用方式为 y=mulawcom(x,mu,V)，其中 mu 是参数 μ 的值，通常取 255；V 指输入量的最大范围。此处把压缩器函数程序列写如下：

```
function y= mulawcom(x,mu,V)
if nargin<3 | V < max(abs(x))
   V=max(abs(x));
end
y = V / log(1 + mu) * log(1 + mu / V * abs(x)) .* sign(x);
```

扩张器函数程序 mulawexp 的输入变元与压缩器相同，建议读者自己列出其子程序。对 A 律的压扩器也可以按(9.2.6) 及其逆函数写出相应的子程序 Alawcom 和 Alawexp，这样就可以很方便地进行以后的比较和演示。

例 9.2.1：画出 μ-律和 A-律的压缩器输入输出曲线。

解：写出如下的 MATLAB 程序 hc921

```
x=0:0.01:1;
y=mulawcom(x,255,1);
y1=Alawcom(x,87.56,1);
plot(x,y,x,y1,':');grid on
legend('\mu 律','A 律')
```

运行此程序的结果见图 9.2.1。请注意 μ-律是和 A-律非常相似的。

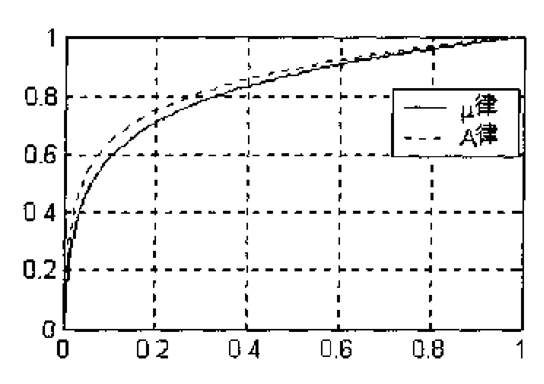
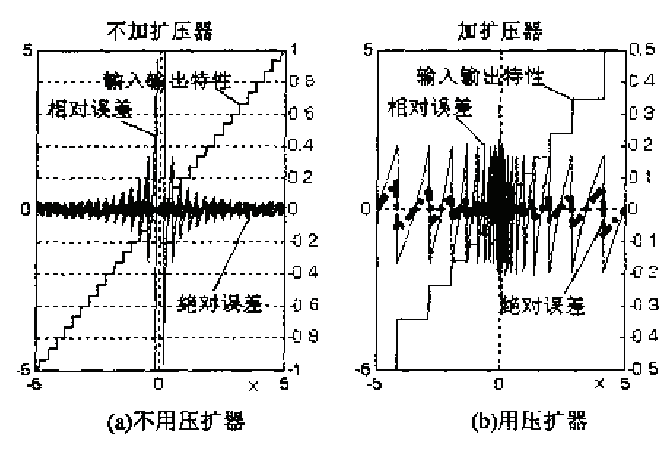


图 9.2.2 μ-律与 A-律非线性压缩器的比较

例 9.2.2：设某 A/D 变换器把最大输入为 5 伏的信号量化为四位二进制（不含符号位），要求用图形描写其输入输出关系，并画出其绝对误差和相对误差的曲线。又若信号像图 9.2.1 那样经过压缩-扩张器，则输入输出关系有何变化？**解**：A/D 变换器的数学模型为绝对量化函数 bqtize，已经在前面得出。利用这个函数，加上求绝对误差和相对误差的语句，可以方便地列出以下的 MATLAB 程序 hc922。

```
x=-5:0.01:5;                              % 输入自变量数组
xq=bqtize(x,4,5);                         % 求量化输出
e=x-xq; er=e./abs(x);                     % 求绝对误差 e 和相对误差 er
plotyy(x,xq,x,er)                         % 用不同的纵坐标画出输出及相对误差曲线
hold on,plot(x,e,'-.','linewidth',3)      %画出绝对误差曲线
```

运行此程序所得曲线见图 9.2.3(a)。



(a)不用压扩器　　(b)用压扩器

图 9.2.3 二进制 A/D 变换输入输出及误差曲线

注意 plotyy 命令生成左右两个不同的纵坐标，分别对应于 *xq* 和 *er*。从图中可以看出，在输入为-5V~5 V 范围内，绝对量化误差 *e* 呈等幅锯齿波形式，其最大值恒定，因而其相对误差 er 在小的输入幅度 *x* 处急骤增大。

如果我们按图 9.2.1 那样，将信号先经过压缩器，再进行 A/D 转换成为数字信号，通过数字通信系统发送出去。在接收端恢复数字信号后，再经过 D/A 转换，通过压扩器输出。用 MATLAB 程序来反映这个框图，可以写成如下的语句：

```
x1=mulawcom(x,255,V);      % 信号经过压缩器
xq=bqtize(x1,4,V);         % 求量化输出
yq=mulawexp(xq,255,V);     % 信号经过扩张器
```

这样的一个系统中，由量化误差 yq-x1 引起的信噪比将比不用压扩器有显著的提高，见

图 9.2.3(b)。从中可以看出它的绝对误差随 x 的增加而增大，其相对误差则在整个输入范围内呈等幅锯齿波形。

9.3 系数量化和运算量化的影响

要用计算机处理信号，就不可避免地要遇到量化问题。这里包括三方面：信号的量化、系统参数的量化以及信号与系统相互作用，也就是运算中的量化。关于信号的量化前面已经讨论过，本节将讨论其他两方面的量化问题。参数量化与数的表示方法有关。现先看一下 MATLAB 中的量化，作为一种通用计算机上的算法语言，它用很多位数和浮点方法来表示数和数的运算，所以它的量化步长很小，精度很高，完全可以当成模拟量来看待，现分析如下。

MATLAB 按二进制双精度格式来表示数，表达一个数需要 8B（字节），也就是 64 bit（二进制位）。按照 IEEE 标准，双精度浮点数 η 用下式表示：

$$\eta = (-1)^S \cdot M \cdot 2^E \tag{9.3.1}$$

其中 M 是一个小于大于 1/2 的二进制分数，称为尾数，占用 52 bit 表示：

$$1/2 < M < 1 \tag{9.3.2}$$

而指数 E 是一个带符号的二进制整数，占 11 bit，总共可表示 2048 个整数，即可以表示从-1023 到+1024 的数集，它决定了数的动态范围。数的正负号反映在 S 上，它只占 1 bit，总计 1+52+11=64 bit。

浮点数的量化步长可以代表它的相对精度，它是由 M 的位数决定的。52bit 二进制数的量化步长是 $2^{-52}=2.2204\times10^{-16}$。该数的动态范围则取决于指数部分。因为 $2^{-1023}\approx10^{-307}$ 及 $2^{+1024}\approx10^{+308}$。所以 MATLAB 中数的动态范围为 $2.2251\times10^{-308}\sim1.7977\times10^{+308}$。在 MATLAB 命令窗中，键入 eps, realmin, realmax，系统就会给出上面的几个数。在做浮点运算时，除非出现大数相减，相对误差可以基本不变。由于多次运算造成误差的积累，MATLAB 中的大部分运算结果的误差比 eps 要大一些。因此正常情况下（即系统不给出警告时），计算结果至少可以有 12 位十进制有效数字的可信度。

当要把用 MATLAB 设计的滤波器付诸实现，特别是用于实时系统时，要采用嵌入式的数字信号处理芯片，通常它必须采用较短的字长（8 位、16 位或 32 位），用定点的方法（运算速度快）来表示数并进行数的运算。因此在工程实践中，绝不可以盲目轻信 MATLAB 中计算出的结果，而必须考虑实际系统的有效字长对结果进行复核和修正。当然实际系统的位数愈高，则它将愈好地接近 MATLAB 计算的结果。

系数和信号的不同在于它可以在一个很大的范围内取值；它的值并非来自 A/D 变换，而是由设计者设定的，各个系数没有必要用同样的量化步长。通常应该尽量利用系统的硬件资源（字长），把各个系数设置得精确一些，因此可以使各系数的相对量化误差为最小。本节将先给出系数相对量化的算式，然后考虑滤波器系数的量化误差对滤波器性能的影响。这个问题涉及到误差分析，而误差分析涉及数的表示方法（定点还是浮点）；也涉及量化的方法（是截断还是四舍五入）；还涉及几个量化后的数进行何种运算（相加还是相减、相乘还是相除）。因为量化是一个非线性问题，很难有封闭形式的解，通常要用大量仿真和统计的方法来分析。在 MATLAB 中就提供了定点运算模块库（Fix-Point Blockset）进行仿真。作为教科书，只能用一些数字例子说明在进行滤波器结构选择时应考虑的原则。

9.3.1　系数相对量化的数学算式和子程序

仍以第 1 章的 1.1 节中的数据为例，xd=[3.14159,0.38455,54.3753,−0.134273]，看这个数据的相对量化的结果。我们都熟悉其十进制模式，就是有效位的长度。在 MATLAB 中有一条命令来实现这个功能，其调用格式为 y=chop(x,N)。其中 N 为有效位长度。例如键入 xdr2=chop(xd,2)和 xdr3=chop(xd,3)，得到的结果分别为

```
xdr2 =    3.1000    0.3800   54.0000   -0.1340
xdr3 =    3.1400    0.3850   54.4000   -0.1343
```

所以 xdr2 与 xd 之间的最大误差为十进制第三位处的±5。根据系数绝对值的最高位大小不同，其最大相对误差的范围为±(0.5~5)%。这个误差的波动范围很大。在计算机中是二进制，所以量化时相对误差的波动范围小的多。可以把上节的二进制量化子程序 bqtize 做一点修改，将其中的量化步长 deltax 不取成标量，而取成数组。就是根据 x 中每个元素的幅值（由其二的幂次 m=$\log_2$x 表示），除以(2⋯N),得到该元素的量化步长。设要把 x 量化为 N 位二进制数，可写出下列二进制相对量化子程序 brqtize.m。

```
function y=brqtize(x,N)
ax=abs(x);                          % 去掉符号
m=ceil(log(ax)/log(2));             % 确定 x 的幅值占二进制 m 位
deltax=2.^(m-N);                    % 求出量化步长，它是一个与 x 同长的数组
xint=round(ax./deltax);             %将 x 除以量化单位，再取整
% 若将 round 改成 fix,则属于截断量化
y=sign(x).*xint.*deltax;            % 再将此整数乘以 2^(m-N),恢复量化后的原值
```

程序中采用了

$$m=\lceil \log_2 x\rceil=\lceil \log(x)/\log(2)\rceil \tag{9.3.1}$$

来确定数 x 的最高二进制位数 m，其中$\lceil \cdot \rceil$表示向上取整。得到 m 后，就可以知道该数的最小量化单位 $\Delta=2^{(m-N)}$，参看(1.1.1)式就可以理解此子程序中后两条语句。这个子程序的好处一是适用于 x 为数组的情况，所以不用一个个系数去量化，特别方便；二是对所有的系数，其相对量化误差都以各个系数本身的绝对值来计算，最大相对误差都是 2^-N。因此它可以对滤波器系数向量实现统一的量化，也可以直接放在调用滤波器系数向量的函数中。例如，仍取前面给出的 xd，将它量化至 4 位二进制，键入 xdq=brqtize(xd,4)，得到

```
xdq =    3.2500    0.3750   56.0000   -0.1406
```

求其相对误差(xdq-xd)./abs(xd)，得到

```
ans =    0.0345   -0.0248    0.0299   -0.0473
```

可见相对误差的绝对值比较均匀，且都在 2^(−4)=0.0625 以下。

9.3.2　系数量化对 FIR 滤波器的影响举例

例 9.3.1：设一个四阶 FIR 滤波器，其传递函数为：

$$H(z)=0.069735+0.388726z^{-1}+0.360530z^{-2}+0.388726z^{-3}+0.069735z^{-4} \tag{9.3.2}$$

算出它的级联结构，比较在两种结构下系数量化为四位二进制对频率响应的影响。

解：先求出其级联结构参数：

```
b=[0.069735,0.388726,0.360530, 0.388726, 0.069735];
[sos,G]=tf2sos(b,1);
```

得到
```
sos =1.0000    4.9315    1.0000    1.0000         0         0
```

```
        1.0000    0.6428    1.0000    1.0000       0       0
    G =   0.0697
```

如果把系数 b 量化为四位二进制，称为 bq，它的值可由键入下面语句

```
bq=brqtize(b,4)
```

得到

```
bq =   0.0703   0.3750   0.3750   0.3750   0.0703
```

如果滤波器由级联结构实现，则应对它的参数进行量化 sosq=brqtize(sos,4)，Gq=brqtize(G,4)。

得到 sosq =

```
        1.0000    5.0000    1.0000    1.0000      NaN      NaN
        1.0000    0.6250    1.0000    1.0000      NaN      NaN
   Gq = 0.0703
```

可以比较以下三种情况的幅频特性。第一种用原来的无限精度系数，第二种用对直接结构量化了的系数，第三种用对级联结构量化了的系数，程序为：

```
[H1,w1]=freqz(b,1);plot(w1,abs(H1)),hold on
[H2,w2]=freqz(bq,1);plot(w2,abs(H2),'-.'),
[H3,w3]=freqz(conv(sosq(1,1:3),sosq(2,1:3)),1);plot(w3,Gq*abs(H3),':'),
legend('理想系数','直接量化','级联量化')
```

得出的幅频特性如图 9.3.1 所示。以理想情况作为标准，可以看出，在本例中用级联结构时，系数量化对系统幅频特性的影响较直接结构小。

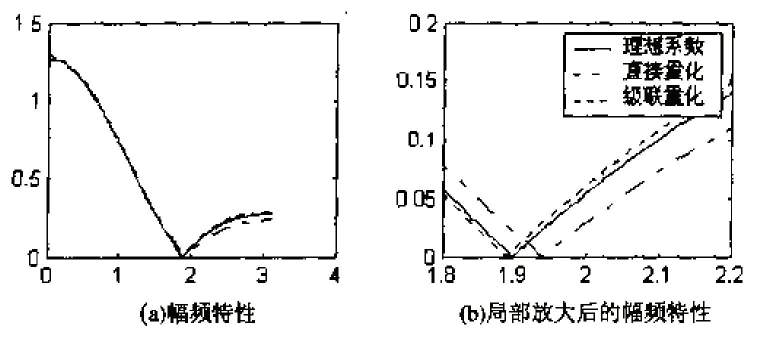


图 9.3.1　不同 FIR 滤波器结构系数量化对幅频特性的影响

9.3.3　系数量化对 IIR 滤波器的影响举例

IIR 滤波器存在着递归计算的问题，它的系数误差可能在循环计算中不断扩散，因此，一般地说，对系数误差的敏感程度比 FIR 滤波器严重。特别是当 IIR 滤波器的极点靠近 z 平面上单位圆时，幅频特性通常要出现很大的峰值。系数的量化误差造成的影响很难用解析公式表示，只能通过实例来计算和观察。

例 9.3.2： 已知一 IIR 滤波器具有如下传递函数：

$$H(z)=\frac{z^{-1}+0.4z^{-2}-0.03z^{-3}+0.232z^{-4}}{1-2.4760z^{-1}+2.9541z^{-2}-1.9631z^{-3}+0.4900z^{-4}}$$

$$=\frac{z^{-1}(1+0.8z^{-1})(1-0.4z^{-1}+0.29z^{-2})}{(1-0.9900z^{-1})(1-0.4999z^{-1})(1-(0.4930+0.8643\mathrm{j})z^{-1})(1-(0.4930-0.8643\mathrm{j})z^{-1})}$$

或
$$H(z)=\frac{z^{-1}(1+0.8z^{-1})(1-0.4z^{-1}+0.29z^{-2})}{(1-1.4900z^{-1}+0.4950z^{-1})(1-0.9860z^{-1}+0.9900z^{-2})} \tag{9.3.3}$$

若将其系数经过 7 位二进制舍入量化，分析其极点的变化。

解：此系统在系数量化前、后的极点可由下列 MATLAB 程序 hc932 求得

```
b=[1,0.4,-0.03,0.232];                      % 分子系数向量
a=[1, -2.4760, 2.9541,-1.9631,0.4900];      % 分母系数向量
aq=brqtize(a,7), brq=brqtize(b,7);          % 量化后的分母、分子系数向量
ra=(aq-a)./abs(a),                          % 量化后的分母系数的相对误差 ra
p=roots(a),pq=roots(aq),                    % 分母系数量化前、后的极点
rp=abs(pq-p)./abs(p)                        % 量化前、后极点的相对误差 rp
zplane(z,p), hold on,                       % 画出量化前零、极点位置(o 和 x)
plot(pq,'*'),plot(zq,'+')                   % 画出量化后零、极点位置(+和*)
```

程序运行的结果如下：

```
aq =    1.0000   -2.4688    2.9688   -1.9688    0.4883
ra =         0    0.0029    0.0050   -0.0029   -0.0035
```

可以看出，系数量化的相对误差 ra 都不到 0.5%。下面看看极点位置的变化。

```
p =  0.4930 + 0.8643i
     0.4930 - 0.8643i
     0.9900
     0.4999
pq = 0.5037 + 0.8730i
     0.5037 - 0.8730i
     0.9613
     0.5000
```

极点位置的相对变化为：

```
rp = 0.0139
     0.0139
     0.0291
     0.0001
```

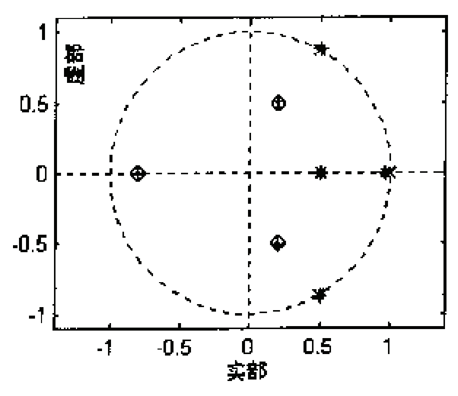


图 9.3.2　例 9.3.2 中系数量化引起的极点移位

可见极点的最大相对误差达到了 3%。问题不单是数量的变化，算一下量化后极点的模 abs(pq)，可以发现前两个共轭复根的模为 1.0079。就是说它们从单位圆内移到单位圆外，使本来稳定的系统变成了不稳定滤波器。量化前后滤波器极点位置的变化见图 9.3.2。量化前的极点用“x”表示，而量化后的极点用“*”表示。

如果以级联方式实现这个系统，着重看这个系统的分母部分，那就可以分别对三个因式进行量化，或者把两个实根合并成为一个二次项，对两个二次项的系数进行量化。程序为：

```
a1=[1,-1.49,0.495];a2=[1,-0.986,0.99];     % 两个二次项的系数向量
a1q=brqtize(a1,7); a2q=brqtize(a2,7);      % 两个系数向量的量化结果
p1q=roots(a1q), p2q=roots(a2q)             % 求系数向量的量化后的根
```

两个量化后的系数向量为：

```
a1q =   1.0000   -0.9844   0.9922
a2q =   1.0000   -1.4844   0.4961
```

它们所对应的极点为：

```
p2q = 0.4922 + 0.8660i
      0.4922 - 0.8660i
p1q = 0.9762
      0.5082
```

这四个极点都在单位圆内部，因此不会造成滤波器的不稳定性问题。

从这个例子，又一次看到，滤波器的级联结构对系数量化的敏感性较之直接结构要小，这里的例子还只是四阶。从多项式理论可以知道，高次多项式系数的微小变化往往会引起根的显著变动。因此，阶次高的滤波器不宜采用直接结构。

系数量化还会造成其他一些问题，如引起极限环振荡等，这些问题一般出现在量化位数较小的单片式嵌入数字系统中。现在，即使是单片机，只要用在信号处理中，位数也都不小于 16 位。所以这类问题已经不很突出了，而且研究这类问题还没有完美的理论，主要靠针对具体问题的仿真计算来判别量化的影响。所以，我们的讨论也到此为止。

9.3.4　运算量化的影响

信号运算的细致过程可以用信号流图来表示。其中主要的运算就是加法、乘法和时延。量化主要体现在加法和乘法运算中。加法器和乘法器如果工作正常（没有溢出），应该能够保持运算的相对精度。

运算精度的规律如下：两个数相加，输出的有效数位与输入大数的有效数位同；两数相减，输出的相对误差将增大，增大的倍数等于输入大数与差之比；两数相乘，输出的有效数位等于输入两数的最低有效数位。所以只要避免两个数量相近的大数的相减，并使运算器的位数与输入数据的位数相匹配，运算器的量化误差与输入数据的量化误差数量级相同。

两个运算数不外乎信号和滤波器系数，只要知道它们的量化方法和量化位数，又知道它们运算结果的量化位数，就可以求出考虑量化后的运算结果，进行比较。

以两个信号相乘后再相加为例，如果原来的数学关系式为

$$y = k_1x_1 + k_2x_2 \tag{9.3.4}$$

把信号 x_1 和 x_2 按此系统中信号量化的规则量化为 xq_1 和 xq_2，又把系数 k_1 和 k_2 按此系统中信系数量化的规则量化为 kq_1 和 kq_2，则其运算式可写成

$$y_o = kq_1xq_1 + kq_2xq_2 \tag{9.3.5}$$

再对 y_o 按系统中运算器和运算结果规定的量化规则进行量化，就可以得到量化的输出

$$yq = Q[y_o] \tag{9.3.6}$$

由于量化的规则无非是定点或浮点，量化的位数又很容易确定，我们已经给出了所有必需的子程序 bqtize 和 brqtize，因此，用 MATLAB 实现(9.3.5)和 (9.3.6)应该是非常简便的。既然对信号流图中的一个细胞，可以进行量化的仿真计算，整个信号流图的量化仿真计算应该也是不困难的。比如在系统中 $x1$ 和 $x2$ 用的是 8 位带正负号定点量化，$k1$ 和 $k2$ 是 8 位不带正负号浮点量化，yo 是 8 位带正负号浮点量化，则 MATLAB 程序 hc933 为

```
xq1=bqtize(x1,7,absxmax);    xq2=bqtize(x2,7,absxmax);
kq1=brqtize(k1,8); kq2=brqtize(k2,8);
yo=kq1*xq1+kq2*xq2;
yq=brqtize(yo,7);
y=kqi*xq1+kq2*xq2; e=yq-y
```

重要的是要真正弄清楚系统实际的软硬件结构。因为实现一种算法，可能有无数种物理结构，不同的结构对量化的敏感程度都不相同，不能用我们习惯的无量化的数学等效模型去思考处理量化问题。一定要使仿真用的信号流图真正符合实际的系统模型，才能得到有意义的结论。

在某些条件下，一个稳定的滤波器还可能因为量化效应而在无输入的情况下产生持续的微幅振荡。这是一种非线性振荡，称为“极限环”。它的振荡幅度与量化步长相近，所以在量化位数较多的情况下，可以不予考虑。

加法和乘法器的问题一般不在于量化，而在于发生溢出。溢出后如果不加特殊处理，加法器的输出会从最大值跳变到负的最大值，这种情况有时会引起大范围振荡，彻底破坏滤波器的工作，那时，上面的量化模型全都无效了，所以这是绝不允许发生的。要避免这种现象发生，除了对输入量的动态范围进行控制之外，应该把加法器设置成溢出后进入饱和的工作状态，不让它在溢出后出现正负号的突跳。

9.4　数字信号处理在双音拨号系统中的应用

9.4.1　双音多频拨号

双音多频(Dual Tone Multi Frequency，DTMF)是用按键进行电话拨号的制式。它不单用在电话中，还可以用于传输十进制数据的其他通信系统中。DTMF 也广泛应用于电子邮件和银行系统。在这些系统中用户可从电话发送 DTMF 信号来选择语音菜单进行操作。

在第 1 章的 1.2 节中已经对它的功能进行了描述，读者在日常生活中也都有了使用的体验。现在要看看如何利用前面学过的理论来解决这个问题。

在 DTMF 通信系统中，高频音与低频音的一个组合表示 0 到 9 中一个特定的十进制数字，或者字符*和#。按图 1.2.1 的分配方法，四个低频频率表示四行，四个高频频率表示四列，两者的组合共可提供 4×4＝16 个字符。第四列的最高频率目前暂时不用，因此可以提供 12 个字符，而其他四个字符保留以备扩展使用。

DTMF 通信系统是一个很典型的小型信号处理系统，它既有模拟信号的生成和传输部分，这要用到 D/A 转换；又有把它转为数字信号（这要用到 A/D 转换）并进行数字处理的部分；而且为了提高系统的检测速度和降低成本，还开发了一种特殊的 DFT 算法，称为 Goertzel 算法；这种算法在国外的几乎每一本数字信号处理的教材上都要介绍，说明了人们对它的重视。这种算法既可以用硬件（专用芯片）也可以用软件实现，所以 DTMF 系统的设计问题是理论与工程相结合的一个很好的典范。

DTMF 音频可以用计算法或查表法产生。用计算方法得到正弦波形的缺点是要占一些运算时间；查表法的速度较快，缺点是要占一定的存储空间；两个正弦波的数字样本按比例相加在一起。因为采样频率是 8 KHz，硬件必须每 125 ms 输出一个样本。将这个叠合信号送到 D/A 变换器变换成模拟音频信号，通过电话线路传送到交换机。

在接收端，将收到的模拟音频信号进行 A/D 变换，恢复为数字信号，然后检测其中的音频频谱来确定所发送的数字。检测算法可以是用 FFT 算法的 DFT，或是用一组滤波器来提取所需频率。当要检测的音频数目比较少时，用滤波器组实现更节省硬件。下面要叙述如何用 Goertzel 算法实现八个调谐滤波器。

回想在第 3 章讨论的 N 点数据序列 $x(n)$的 DFT 为：

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk},\quad k=0,1,\cdots,N-1 \tag{9.4.1}$$

如果用 FFT 算法实现该 DFT 计算，计算量(复数乘法和加法)是 $N\log_2 N$。这样可立即得到 DFT 的所有 N 个值，但至少要 N 个存储器。然而，如果只希望计算 DFT 的 K 个点，而 $K\ll N$，则直接计算可以节省很多内存，因为只需要八个频点上的频谱分量。下面介绍的 Goertzel 算法就属于后者，它实质上是计算 DFT 的一种线性滤波方法。

9.4.2　Goertzel 算法

Goertzel 算法利用相位因子$\{W_n^k\}$的周期性，使我们能将 DFT 运算表示为线性滤波运算，由于 $W_N^{-kN}=1$，可用该因子去乘 DFT，则

$$X(k)=W_N^{-kN}\cdot X(k)=\sum_{m=0}^{N-1}x(m)W_N^{-k(N-m)}\quad k=0,1,\cdots,N-1 \tag{9.4.2}$$

注意到，(9.4.2)式就是卷积形式。的确，如果定义序列 $y_k(n)$为

$$y_k(n)=\sum_{m=0}^{N-1}x(m)W_N^{-k(n-m)}=x(n)\otimes W_N^{-k(n)} \tag{9.4.3}$$

可见，$y_k(n)$可以看成两个序列的卷积。一个是长度为 N 的有限长输入序列 $x(n)$，另一个则是具有如(9.4.4)式所示的单位脉冲响应的滤波器。

$$h_k(n)=W_N^{-kn}\mu(n) \tag{9.4.4}$$

该滤波器在 $n=N$ 点的输出就是 DFT 在频点 $\omega_k=2\pi k/N$ 的值，即

$$X(k)=y_k(n)\big|_{n=N} \tag{9.4.5}$$

可通过比较式(9.4.2)和式(9.4.3)来验证上式。

单位脉冲响应为 $h_k(n)$的滤波器的系统函数就是(9.4.4)式的 z 变换，容易求出为

$$H_k(k)=\frac{1}{1-W_N^{-k}z^{-1}} \tag{9.4.6}$$

这个滤波器只有一个位于单位圆上频率为 $\omega_k=2\pi k/N$ 处的极点。因此，可使输入数据块通过 N 个并行的单极点滤波器(谐振器)组来计算全部 DFT。Goertzel 算法的好处不在于节省时间，而在于节省空间。如果只需要 K 个 DFT 样本，可以只用 K 个并行的单极点滤波器来分别计算这 K 个样本，其中每个滤波器有一个位于 DFT 相应频率的极点，这就可以大大节省硬软件资源。

由(9.4.6)式给出的滤波器差分方程，还可以用迭代方法计算 $y_k(n)$，而不用(9.4.2)式那样的卷积方法，硬件资源可以更加节省。因为

$$y_k(n)=W_N^{-k}y_k(n-1)+x(n),\qquad y(-1)=0 \tag{9.4.7}$$

预期的输出为 $X(k)=y_k(N)$。为了执行该计算，可以只算一次相位因子 W_N^{-k}，将其存储起

来。递推的框图如图 9.4.1(a)所示。

(9.4.7)式中包含一次复数运算，若将具有一对复共轭极点的谐振器 W_N^{-k} 和 W_N^k 组合在一起，就可避免其中的复数乘法运算，这样就导出具有如下形式系统函数的双极点滤波器:

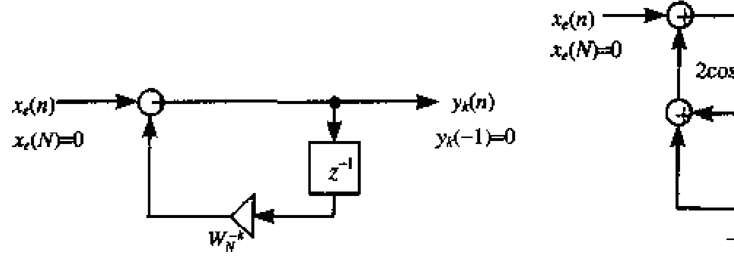
$$H_k(k)=\frac{1-W_N^{-k}z^{-1}}{1-2\cos(2\pi k/N)z^{-1}+z^{-2}}=\frac{Z[v_k(n)]}{Z[x(n)]}\cdot\frac{Z[y_k(n)]}{Z[v_k(n)]} \tag{9.4.8}$$

(9.4.8)式可以用两个差分方程构成的方程组表示为:

$$v_k(n)=2\cos\frac{2\pi k}{N}v_k(n-1)-v_k(n-2)+x(n) \tag{9.4.9}$$

$$y_k(n)=v_k(n)-W_N^{-k}v_k(n-1) \tag{9.4.10}$$

初始条件为 $v_k(-1)=v_k(-2)=0$，这就是 Goertzel 算法的二阶实数算法，其结构图如图 9.4.1(b)所示。(9.4.9)中的递推关系对 $n=0$，$1\cdots N$ 重复 $N+1$ 次，每次计算只需要计算一次实数乘和两次实数加，而带有复数运算的方程(9.4.10)仅在 $n=N$ 时刻计算一次。所以，对实数序列 $x(n)$，由于对称性，用这种算法求出 $X(k)$和 $X(N-k)$的值只需要 N 次实数乘法和一次复数乘法运算。



(a) 计算单点 DFT 的递推框图　　(b) 计算 DFT 的双极点谐振器实现

图 9.4.1　用 Goertzel 算法实现 DFT 计算的示意图

注：其实在 DTMF 解码器中，不需要计算复数值 $X(k)$，只要求出幅度值$|X(k)|$或幅度平方值$|X(k)|^2$就够了。所以，计算涉及分子项(滤波器计算的前向部分)的 DFT 值的最后一步可以简化，有

$$\begin{aligned}|X(k)|^2&=|y_{k^2}(N)|^2=|v_k(N)-W_N^k v_k(N-1)|^2\\&=v_k^2(N)+v_k^2(N-1)-2\cos\left(\frac{2k\pi}{N}\right)v_k(N)v_k(N-1)\end{aligned} \tag{9.4.11}$$

于是 DTMF 解码器中就完全避免了复数运算。

现在能够用 Goertzel 算法实现 DTMF 解码器了。由于有八种可能的音频要检测，所以需要八个(9.4.8)式所给出的滤波器，每个滤波器调谐到这八个频率之一上。下面就是按(9.4.11)式编成的 Goertzel 算法子程序 gfft。它根据输入序列 x 和指定的 DFT 样本的序号 k 计算待求的 DFT 样本 X。其核心语句只有四句，它综合了前面的一系列公式。

```
function X=gfft(x,k)
% 用Goertzel 算法计算序号为 k 的 DFT 样本
N=length(x);x1=[x,0];                % (9.4.9)递推要 N+1 次，故把输入序列长度加 1
d1 = 2*cos(2*pi*k/N);                % 二阶滤波环节中间项系数
v = filter(1,[1,-d1,1],x1);          % 用滤波函数实现卷积
W = exp(-i*2*pi*k/N);                % 为下一步计算求 W
X = v(N+1) - W*v(N);                 % 最后求出第 k 个 DFT 样本
```

9.4.3 检测 DTMF 信号的 DFT 参数选择

DTMF 信号的参数选择要考虑多方面的因素。首先电话数字化的采样频率已知为 8 kHz，实际上为了抑制语音干扰，检测系统除了规定的八个频率之外，还要检测它们的二次倍频处的 DFT 幅度。这些二倍频处幅度应该很小。如果基频和倍频分量同时都大，那就可能是外来声音的干扰。所以在选择基频时已考虑到把它们的八个倍频都放在奈奎斯特频率 4000 Hz 的范围以内，如表 9.4.1 所示。这个双频模拟信号在电话线上传输到接收端（总机），总机用 8 kHz 将它采样为离散时间信号，并判断它所代表的数字，以完成接通的任务。如果序列的长度为 N，则对该序列进行 DFT 所得样本的频率间隔或频率分辨率为 D=8000/N。对应于频率 F_k 的样本序号为：

$$k=F_k / D=Fk * N / 8000$$

现在要分析一下应取的样本长度 N。对于按键的最短时间要求是 40 ms。因此对每个十进制数，至多只能用 0.04×8000=320 个样本来分析检测。实际取的样本数 N 要少于这个值，两头留出裕度。N 也不能太小，分辨率应至少保证两个相邻基频序号差 2，否则容易出错。第一和第二基频的差才 73 Hz，要使其差两个样本序号，则其频率分辨率应为 $D\leqslant36.5$，故 N 应该取 200 以上。在 N = 200～300 范围内再做细致选择时，其主要依据是使得这 16 个频率离开其对应的 k 取整数的 DFT 样本位置的误差为最小。按这个标准，人们研究的结论是：对于已选定的这 8 个频率，N 取 205 是最好的，详细数据见表 9.4.1。

表 9.4.1 各个拨号频率及 N=205 时其对应的 DFT 序号

8 个基频 Hz	准确 k 值	最近整数 k 值	绝对误差	二次谐波频率(Hz)	准确 k 值	最近整数 k 值	绝对误差
697	17.861	18	0.139	1394	35.024	35	0.024
770	19.531	20	0.269	1540	38.692	39	0.308
852	21.833	22	0.167	1704	42.813	43	0.187
941	24.113	24	0.113	1882	47.285	47	0.285
1209	30.981	31	0.019	2418	60.752	61	0.248
1336	34.235	34	0.235	2672	67.134	67	0.134
1477	37.848	38	0.152	2954	74.219	74	0.219
1633	41.846	42	0.154	3266	82.058	82	0.058

现在用一个 MATLAB 程序来演示 DTMF 双频拨号的全过程。首先程序要求用户输入一个电话号码或字符，然后由软件根据这个号码查出它对应的两个频率，并生成相应的双频信号 $x(n)$。这是在发送端的过程，让它产生声音作为标志。在接收端，程序对收到的双频信号进行截取并做傅里叶变换，求出它在八个规定频率上的样本幅度。在行向和列向分别取出满足规定电平的序号，它们也就是两个下标。再由下标找到它的 ASCII 码和字符。为了简化，程序中没有设置抗语音干扰的功能，没有检测倍频频率上的幅度。

此程序名为 hc941，大体分为三段。第一段是双频信号的生成，以产生双频声音结尾；第二段是接收端信号的八个基频 DFT 计算，以八个基频 DFT 幅度分布图结尾；第三段是根据基频的两个峰值，查找和显示输入字符。各段之间用 pause 命令隔开，全部列写如下：

```
% DTMF 检测程序hc941
%
d=input('键入一位电话号码= ','s');                    % 输入一个号码字符
symbol=abs(d);                                        % 求它的ASCII码
tm=[49,50,51,65;52,53,54,66;55,56,57,67;42,48,35,68];  % 16个ASCII码
for p=1:4;
   for q=1:4;
      if tm(p,q)==abs(d); break,end                   % 检测码相符的列号q
   end
      if tm(p,q)==abs(d); break,end                   % 检测码相符的行号p
end
f1=[697,770,852,941];                                 % 行频率向量
f2=[1209,1336,1477,1633];                             % 列频率向量
n=0:2040;                                             % 为了发声,加长序列
x = sin(2*pi*n*f1(p)/8000) + sin(2*pi*n*f2(q)/8000);% 构成双频信号
sound(x);                                             % 发出声音
disp('双频信号已经生成并发出'),pause
% 接收检测端的程序
N=205;k = [18 20 22 24 31 34 38 42];                  % 要求的DFT样本序号
for m = 1:8;
   X(m)=gfft(x(1:205),k(m));                          %用自编Goertzel算法子程序计算八点DFT样本
end
val = abs(X);                                         % 列出八点DFT向量
stem(k,val,'.');grid;xlabel('k');ylabel('|X(k)|')     % 画出DFT(k)幅度
set(gcf,'color','w')
shg,disp('图上显示的是检测到的八个近似基频的DFT幅度');pause
limit = 80;                                           %规定检测门限
for s=5:8;
   if val(s) > limit, break, end                      % 查找列号
end
for r=1:4;
   if val(r) > limit, break, end                      % 查找行号
end
disp(['接收端检测到的号码为',setstr(tm(r,s-4))])       % 显示接收到的字符
```

程序运行时,用户先按提示输入电话按键集中的一个字符,例如7。程序即根据它的ASCII码在键盘矩阵tm中找到与它对应的行号和列号,分别找到相应的行基频和列基频,生成双频信号发出。在接收端用Goertzel算法求此信号的八个基频的DFT幅度val,画出分布图如图9.4.2所示。然后分别检测出它在下标[1:4]和[5:8]两个区间内的峰值下标。再根据这两个下标所对应的行号和列号,查出它对应的字符;这个字符在此情况下也应为7。

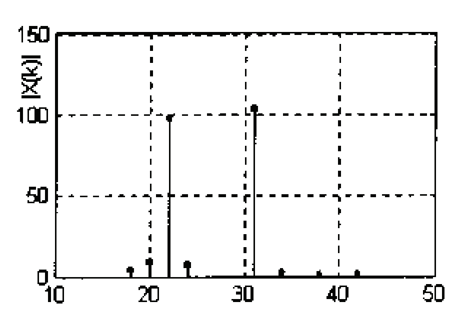


图9.4.2 接收端双频信号在八个近似基频上的DFT幅度

在信号处理工具箱版本5.1中,也给出了Goertzel算法的子程序goertzel.m。它的调用格式与gfft相似,不同之处在于第二变元k可以是数组,所以编程时可以省去一个for循环。另外k指的是fft的下标,所以它等于频率的fft编号加1。因此在程序hc941中,将有下画线的三行for循环语句换成语句

```
X=goertzel(x(1:N),k-1);    % 用工具箱函数 Goertzel 算法计算八点 DFT 样本
```

即得到程序 hc941a，同样能够正确运行。不过工具箱中的这个函数程序有 100 多行，不像本书的 gfft 子程序那么容易读懂，所以讲课时还是要介绍 gfft，而在调用时应尽量用工具箱函数。

9.5 正余弦信号的谱分析

数字信号处理方法的一个重要用途是在离散时间域中确定一个连续时间信号的频谱，通常称为频谱分析，更具体地说，它也包括确定能量谱或功率谱。数字频谱分析可以应用在很广阔领域，频谱分析方法是基于以下的观测，如果连续时间信号 $g_a(t)$是限带的，那么它的离散时间等效物 $g(n)$应当能根据(4.5.1)式给出 $g_a(t)$频谱的一个很近似的估计两者之间只差一个带数因子 T。然而，在大多数情况下，$g_a(t)$是在$-\infty<t<\infty$范围内定义的，因此 $g(n)$也就定义在$-\infty<t<\infty$的无限范围内，要估计一个无限长信号的频谱是不可能的。实用的方法是：先让模拟连续信号 $g_a(t)$通过一个抗混叠的模拟滤波器，然后把它采样成一个离散序列 $g(n)$。假定反混叠滤波器的设计是正确的，则混叠效应可以忽略，又假设 A/D 变换器的字长足够长，则 A/D 变换中的量化噪声也可忽略。

假定表征正余弦信号的基本参数，如振幅、频率和相位，不随时间改变，则此信号的傅里叶变换 $G(\mathrm{e}^{\mathrm{j}\omega})$ 可以用计算它的 DTFT 得到

$$G(\mathrm{e}^{\mathrm{j}\omega})=\sum_{n=-\infty}^{\infty}g(n)\mathrm{e}^{-\mathrm{j}\omega n} \tag{9.5.1}$$

实际上无限长序列 $g(n)$首先乘以一个长度为 N 的窗函数 $w(n)$，使它变成一个长为 N 的有限序列，$g_1(n)=g(n)w(n)$，对 $g_1(n)$求出的 DTFT $G_1(\mathrm{e}^{\mathrm{j}\omega})$ 应该可以作为原连续模拟信号 $g_a(t)$的频谱估计,然后求出 $G_1(\mathrm{e}^{\mathrm{j}\omega})$ 在 $0\leqslant\omega\leqslant2\pi$ 区间等分为 R 点的离散付立叶变换 DFT。为了保证足够的分辨率，DFT 的长度 R 选得比窗长度 N 大，其方法是在截断了的序列后面补上 $R-N$ 个零。计算采用 FFT 算法。

我们更详细地考察一下上面的方法。这样才能了解它的限制，并正确利用它所得出的结果。特别要分析加窗的效果，以及和由 DFT 样本来估计 DTFT 频率采样值的问题。

在讨论由 $G_1(k)$来估计频谱 $G_1(\mathrm{e}^{\mathrm{j}\omega})$ 和 $G(\mathrm{e}^{\mathrm{j}\omega})$ 时，需要重新探讨一下这些变换和它们所对应的频率之间的关系，R 点的 DFT $G_1(k)$与它的 DTFT $G_1(\mathrm{e}^{\mathrm{j}\omega})$ 的关系为：

$$G_1(k)=G_1(\mathrm{e}^{\mathrm{j}\omega})\Big|_{\omega=2\pi k/R}\qquad 0\leqslant k\leqslant R-1 \tag{9.5.2}$$

归一化的数字角频率 ω_k 和 DFT 样本序号 k 的关系为

$$\omega_k=\frac{2\pi k}{R} \tag{9.5.3}$$

同样地，模拟角频率和 DFT 样本序号 k 的关系为

$$\Omega_k=\frac{2\pi k}{RT} \tag{9.5.4}$$

其中 T 是采样周期。为了能正确地解释以 DFT 为基础的频谱分析结果，首先来考虑单频

率正余弦序列的频域分析。设一个具有数字角频率 ω_0 的余弦信号为：

$$g(n)=\cos(\omega_0 n+\phi) \tag{9.5.5}$$

把这个序列表为：

$$g(n)=\frac{1}{2}\left(e^{j(\omega_0 n+\phi)}+e^{-j(\omega_0 n+\phi)}\right) \tag{9.5.6}$$

查表得知它的 DTFT 为：

$$G(e^{j\omega})=\pi\sum_{l=-\infty}^{\infty}\left[e^{j\phi}\delta(\omega-\omega_0+2\pi l)+e^{-j\phi}\delta(\omega+\omega_0+2\pi l)\right] \tag{9.5.7}$$

因此，它是一个以 2π 为周期的 ω 的周期信号，每个周期中包含两个冲击信号。在 $-\pi\leqslant\omega\leqslant\pi$ 的频率范围内，$\omega=\omega_0$ 处的冲击具有复数幅特性 $\pi e^{j\psi}$，而在 $\omega=-\omega_0$ 处具有 $\pi e^{-j\psi}$ 的复数幅特性。

为了用 DFT 分析 $g(n)$ 的频域特性，取 $g(n)$ 的一个有限长序列

$$g_1(n)=\cos(\omega_0 n+\phi) \qquad 0\leqslant n\leqslant N-1 \tag{9.5.8}$$

假如是一个频率为 10 Hz 而采样频率为 64 Hz 的 32 点序列，则用 MATLAB 表示 g1 并计算它的 DFT 的程序语句如下：

```
N=32; f=10;Fs=64
n=0:N-1;g1=cos(2*pi*f*n/Fs);k=n;
G1=fft(g1,N);stem(k,abs(G1),'.')                    % 画出 DFT 样本点
G=fft(g1,1024);plot(N*[0:1023]/1024,abs(G),'-.')    % 画出 DTFT 曲线做参考
```

计算结果见图 9.5.1(a)。它的 DFT 只有两个点不等于零，位于 k=5 和 k=27 处，经公式(9.5.4)知道，k=5 对应于频率 10 Hz，k=27 对应于频率 54Hz（也就是-10 Hz），这样 DFT 确实正确地分辨了余弦信号的频率。但是这样理想的结果是碰巧得到的，因为我们恰好截取了五个完整的余弦周期(f*N/Fs=5)。如果截取的不是整数周期，情况就不同了。

例如把频率 f 改为 11 Hz，而采样频率仍为 64 Hz 和窗长度仍为 32 点，用同样的程序计算此余弦信号的频谱，则计算结果见图 9.5.1(b)。频谱图上 k=5 和 k= 27 处都有较大的峰值，而其他的点上幅度不再为零。这两个峰值对应的频率是 10 Hz 和 12 Hz，因此可以判断，信号的频谱峰值确实位于两者之间。本来是单一的 11 Hz 频率的能量会分布到许多 DFT 频率上的现象称为频率泄漏，来源于截断效应。

从图 9.5.1 中可以看到，(a)、(b)两个子图的 DTFT 形状是很相像的。两个 DFT 样本点所以有那么大的差别，原因就在于上图的采样点位置正巧都在频谱的零点。实际工程中，输入信号的参数是未知的，而且通常含有丰富的各种频谱，这种理想情况不会出现。为了理解图 9.5.1 中的频谱形状，可以把有限序列 $g_1(n)$ 看做无限序列 $g(n)$ 和长度为 N 矩形窗序列的乘积。两个时域的序列乘积的频谱，应该等于它们的频谱的卷积。

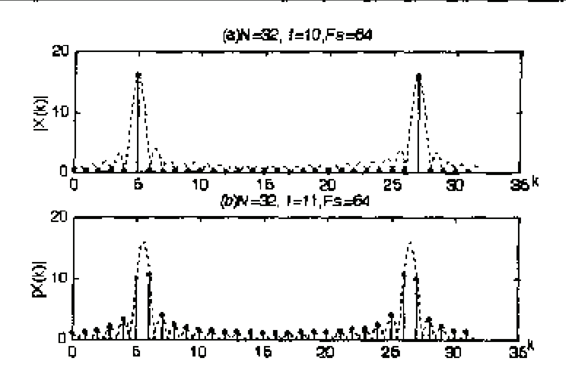


图 9.5.1 单频率有限长正余弦信号的 DFT 样本点和 DTFT 幅频特性曲线

$$X_1(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\varphi})Wd_R(e^{j(\omega-\varphi)})d\varphi \tag{9.5.9}$$

其中

$$Wd_R(e^{j\omega}) = e^{-j\omega(N-1)/2}\frac{\sin(\omega N/2)}{\sin(\omega/2)} \tag{9.5.10}$$

为矩形窗函数的频谱函数。由此算出

$$X_1(e^{j\omega}) = \frac{1}{2}e^{j\phi}Wd_R(e^{j(\omega-\omega_0)}) + \frac{1}{2}e^{-j\phi}Wd_R(e^{j(\omega+\omega_0)}) \tag{9.5.11}$$

上述公式说明，加窗后的序列 $g_1(n)$的 DTFT $G_1(e^{j\omega})$是窗函数 $wd(n)$的 DTFT $Wd_R(e^{j\omega})$ 移频$\pm\omega_0$和加权后的和。对于频率为 11Hz 的序列，它的数字频率为 ω_0=11/64*2π=0.344π，因序列长度 N=32,频率分辨率为 $\Delta\omega=2\pi/N\approx0.0625\pi$。$\omega_0/\Delta\omega$=5.5，因此 ω_0 在 k 为横坐标的图上是在 5 和 6 之间。DFT 是由此长度 32 的矩形窗的频谱，向左右各移动 5.5 后再相加并乘 1/2 而得的。在数字角频率 0~2π(数字下标 k=0~N-1)范围内，存在着两个尖峰，一个在 k=5.5 处，表现为 k=5 和 k=6 之间两处的 DFT，另一个在 k=32-5.5=26.5 处，表现为 k=26 和 k=27 处的两个 DFT。所有其它的 DFT 样本是由窗函数的泄漏引起的 DTFT 的旁瓣所造成。相对旁瓣水平 A_{Sl}(dB)定义为主瓣与最大旁瓣之比，因为矩形窗的旁瓣 A_{Sl} 特别大，所以它所引起的泄漏也特别严重。

当输入信号有一个以上的正余弦分量时，上述问题变得更加严重，现将通过几个案例来加以说明。着重分析窗函数的类型、它的长度 N 和 DFT 的长度 R 对频谱分析的影响。

例 9.5.2：考察 DFT 的长度对双频率信号频谱分析的影响。设待分析的信号为

$$x(n) = 0.5\sin(2\pi f_1 n) + \sin(2\pi f_2 n) \qquad 0\leqslant n\leqslant N-1$$

令两个长度为 16 的正余弦序列的数字频率为 f_1=0.22 及 f_2=0.34。

解：现在计算 $x(n)$在取不同 R 值时的 DFT 值。为此采用下列程序，其中 R、N、f_1、f_2 都是可设定的。

```
N=input('信号长度');                                  %
R=input('DFT 长度');                                  %
fr=input('[f1,f2]=');                                 % 输入两个正弦频率
n=0:N-1;                                              % 设定自变量向量
x=0.5*sin(2*pi*n*fr(1))+sin(2*pi*n*fr(2));            % 两个正弦信号合成
X=fft(x,R);k=0:R-1;                                   % 求其 DFT
stem(k,abs(X),'.');grid on                            % 画图
```

以这几条核心语句为基础，编出程序 hc952a。它取 N=16，取 R 为四个不同值 16,32,64,128，画出的四个 DFT 幅度图如图 9.5.2(a)、(b)、(c)、(d)所示。在图 9.5.2(e)中画出了 x 的 DTFT

幅度曲线$|X(\omega)|$。很明显这些DFT样本值就是其DTFT在相应位置的采样。因为DFT和DTFT的频率轴刻度不同，需要做一个换算：对$k=8$，$\omega=8\times2\pi/16=\pi$，而对$k=15$，$\omega=15\times2\pi/16=1.875\pi$。

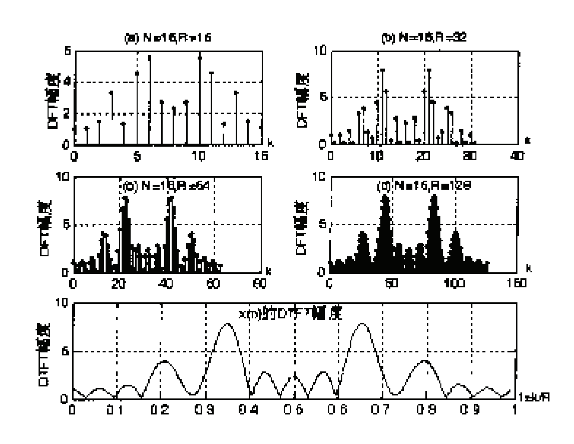


图 9.5.2 双频率 fr=[0.22,0.34] 有限长正余弦信号的DTFT幅频特性曲线

从图9.5.2(a)上很难看出它有两个峰值，因此就要提高它的分辨率。现把DFT的长度R由16增加到32，64和128，这样就得出了图9.5.2(b)、(c)、(d)3个子图。逐渐可以看出它有两个峰值。注意这几个图的横坐标为k，可换算为数字频率$f=\omega/2\pi=k/R$。这样就能确定峰值的位置大体在$f=0.21$和0.35附近,与信号的给定频率有一定误差，这也是截断和泄漏带来的问题。在这图上还可看到一些较小的峰，这时很难判断这些小峰是输入信号里固有的，还是由泄漏所引起的。

这个例子说明，一般说来，增加DFT长度R减小了相邻样本间的频率间距，提高频谱的视在分辨率，因而可以提高样本位置的测定精度。

下面来看窗函数长度N的影响。前面指出，有限长度信号的DTFT等于原无限信号的频谱与窗函数的频谱的卷积。两个频率不同的无限长序列的频谱应该是两对冲击函数，本来应该可以分辨的，但由于和有限长的窗函数的频谱进行卷积，就形成了较宽的频谱图形。矩形窗函数的主瓣宽度为$4\pi/N$，其他所有的窗函数主瓣宽度都大于矩形窗，也都与N成反比。因此增加序列长度N可以有效地提高频谱的实际分辨率。

虽然矩形窗的主瓣宽度最小，但它的旁瓣幅度太大，造成严重的频率泄漏，这会使频谱分析的可靠性和精确性下降。因此，人们宁可选一个没有旁瓣的窗函数，而靠加大N来提高频谱的分辨率。下面的例子说明用Hamming窗对改善分辨率的影响。

例 9.5.3：在上例中若把两个正弦波的频率取得较近，令$fr=[0.22,0.25]$，试问怎样选择FFT参数才能在频谱分析中分辨出这两个分量？

解：要使这两个频率之间隔开一个样本，分辨率至少应达到$\Delta f=0.03/2=0.015$。因为此处的数字频率是对采样频率Fs进行归一化的，即fr的最大值为1。因此总的样本数（也就是N）至少要达到 1/0.015=66。为了分析加窗的影响，只要在程序 hc952a 中增加一句 x1=x.*hamming(N)，然后对$x1$求DFT并画曲线即可。选择不同N、R和窗函数，编成程序hc953，运行此后，可得到图 9.5.3。从此图上可以看出，信号的两个峰值是能够分辨的。但分辨的质量如何，需要放大后才能看清。

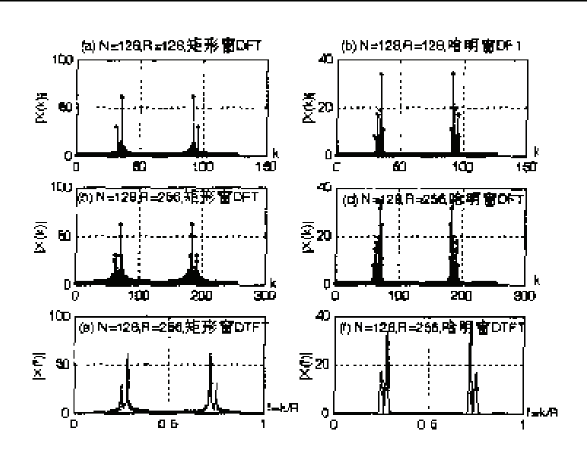


图 9.5.3　双频率 fr=[0.22,0.25]正弦信号的 DFT 和 DTFT 幅频特性曲线

将图 9.5.3 的横坐标放大，可以得到图 9.5.4，左边的(a)、(c)、(e)子图是加矩形窗的结果，右边的(b)、(d)、(f)子图是加哈明窗的结果。可以看出，使用无旁瓣的窗函数得到的频谱函数比较光滑，便于分辨峰值位置和准确的数值。所以在频谱分析仪中都要采取加窗的措施。为了提高实际的分辨率，应该尽量增加信号长度 N 及 DFT 长度 $R(\geqslant N)$，当受到条件限制，不能提高 N 时，单独提高 R 也可以提高视在分辨率，因而有助于测量精度改善。

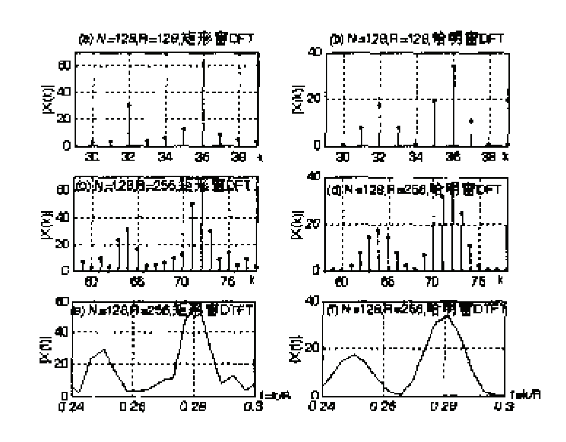


图 9.5.4　把图 9.5.3 的横轴放大后的 DFT 和 DTFT 幅频特性曲线

在工程实践中，人们所遇到的大多数信号，例如语音通信、雷达信号等等，都不是平稳的。在这种情况下，增加取样的长度 N 未必有实际的意义，很多的时候，要求人们进行非平稳信号的频谱分析，于是出现了多种多样的分析方法。第 9.1 节谈到的 sptool 中，在频谱分析的选项时就列出了好多种。这些都要用比较高深的数学工具，在研究生阶段的《高等数字信号处理》课程中才安排学习。

这里简单介绍一种，称为短时间傅里叶变换(Short-Time Fourier Transfom，STFT)方法。序列 $x(n)$的 STFT 定义为：

$$X_{\mathrm{STFT}}(\mathrm{e}^{\mathrm{j}\omega},n)=\sum_{m=-\infty}^{\infty}x(n-m)wd(m)\mathrm{e}^{-\mathrm{j}\omega m} \tag{9.5.12}$$

其中，$wd(m)$是一个适当选择的窗函数序列。实际上这个式子说明，窗函数序列在 $x(n)$ 中选择了一个有限长度序列。在这个较短的时间内，可以近似把非平稳过程当做平稳过程来进行傅里叶分析。对于一个非平稳序列，如果窗函数随时间变化而移动，则得出的频谱也将随时间而变化。对于随时间而变化的频谱，时间轴和频率轴就占了两维，用两维图形无法表

示其幅度了，于是第三维就用颜色或图形的灰度来表示。

例如一个线性调频(Chirp)信号的频谱可以用图 9.5.5 表示。其中横坐标为时间轴，表示窗的位置；纵坐标为频率，向上为频率增加；灰度表示幅特性，灰度最高（全白）意味着幅度最大。向上的白色斜线说明频谱最大值对应的频率随时间的增加向高频端移动。下列程序 hc954 可以给出这个频谱图，STFT 函数 specgram 取的默认输入变元为，FFT 长度 R=256，序列长度 N=R/2，窗函数为汉宁窗。

```
t=0:0.001:2;                          % 1kHz 采样频率，持续 2 秒
x=chirp(t,0,1,150);                   % 从直流开始，在 1 秒时达到 150Hz 的线性调频
subplot(2,1,1),plot(t,x)              % 画出信号曲线
subplot(2,1,2),specgram(x);           % 显示频谱图
```

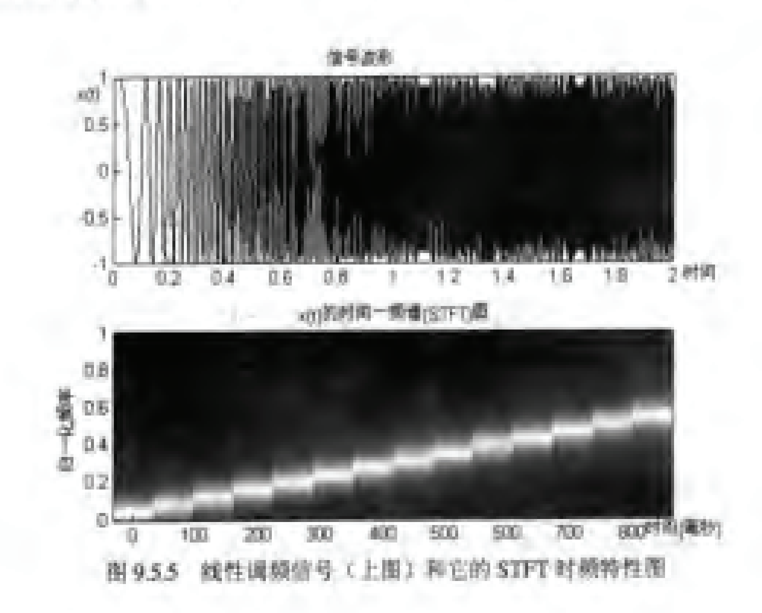


图 9.5.5　线性调频信号（上图）和它的 STFT 时频特性图

9.6　音乐信号处理

由于 CD，DVD 和家庭视听技术的发展，音乐的录制和加工已经愈来愈多地采用数字技术。音乐的录制大致分以下几个步骤：首先在一个隔音的舞台上把乐队中各个乐器的声音分别录在一个多磁道磁带的各个独立磁道上；然后由音响工程师把各个磁道上的信号进行单独的处理，加入特定的声音效果；最后在一个混音系统中把这些信号进行合成，录制在一个立体声的双磁道磁带上。这些特定的声音效果已愈来愈多地靠数字信号处理技术来实现，本节将作一简单的介绍。

9.6.1　时域处理方法

在一个像音乐厅那样的封闭空间中，人们听到的声音包括直接传播声音、一次反射声音和混响(reverberation)等几种成分。一次反射声音主要是由直接传播声音在近距离内的若干直接回声构成，而混响则是由密集的回声形成的。所以在一个隔音的舞台上录制的音乐与音乐厅中录制的效果就不一样，人们听起来会感到“不自然”。这时就需要用数字滤波器来人为地改变录制的信号，通过增加一些回声，使它接近于音乐厅中的效果。

回声可以用延迟单元来生成。直接声音和它的延迟了 R 个周期的单个回声可以用如下的差分方程表示：

$$y(n)=x(n)+\alpha\, x(n-R) \qquad |\alpha|<1 \tag{9.6.1}$$

其中 $\alpha<1$ 表示回声的衰减系数。上述差分方程也可以用传递函数为:

$$H(z)=1+\alpha\, z^{-R} \tag{9.6.2}$$

的 FIR 滤波器来实现，这实际上是一个梳状滤波器，其结构如图 9.6.1(a)。可以用实验来证明这一点。将“大家好”文件中的变量 x 作为输入语声 $x(n)$，让它通过由(9.6.2)式表示的滤波器，把得到的 $y(n)$放音检验。为了得到 0.3s 的延迟和 30%的衰减，滤波器参数 R 选为 $0.3Fs=0.3\times22050=6615$，$\alpha=0.3$。程序 hc961 的语句如下：

```
load dajiahao
y=filter([1,zeros(1,6615),0.3],1,[x;zeros(5000,1)]);
sound(y,22050)
```

注意由 filter 函数产生的输出序列长度与输入序列相同。为了使输出时间足够长，以便包含回声的位置向量，程序中用补零的方法把输入 x(列向量)加长了 5000 个样本。

为了生成间隔为 R 个周期的多重回声，（9.6.2）式应当改为

$$H(z)=1+\alpha\cdot z^{-R}+\alpha^{2}\cdot z^{-2R}+\cdots+\alpha^{N-1}\cdot z^{-(N-1)R}=\frac{1-\alpha^{N}\cdot z^{-NR}}{1-\alpha\cdot z^{-R}} \tag{9.6.3}$$

它的结构如图 9.6.1(b)。如果反射的次数很多，$\alpha^N\to0$，则多重回声可以用一个 IIR 滤波器来实现。其传递函数为

$$H(z)=\frac{1}{1-\alpha\cdot z^{-R}} \qquad |\alpha|<1 \tag{9.6.4}$$

在上述 MATLAB 程序第二行中，把滤波器的分子分母系数调换一下，同时把 α 取大一些，又把 x 加得更长，以便得到更多多重回声，写成以下求合成声音 y1 的语句。

```
y1=filter(1,[1,zeros(1,6615),0.8],[x;zeros(50000,1)]);
sound(y1)
```

这样就可以检验其回声效果。

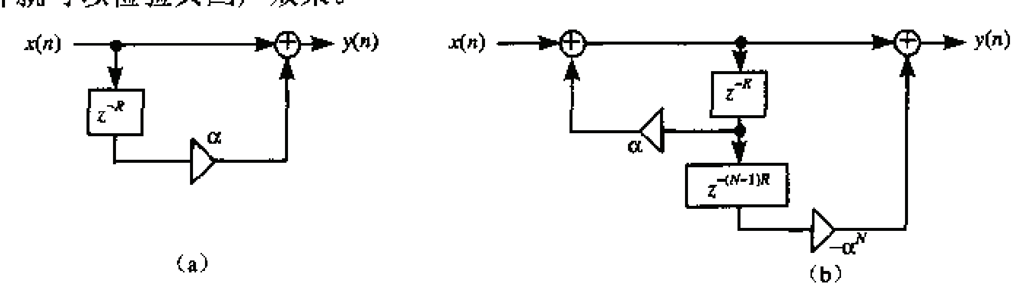


图 9.6.1　音乐回声信号仿真的两种滤波器结构

不过这样简单的梳状滤波器其实还无法满足音乐处理的要求。首先是因为这种滤波器的幅频特性不是常数，对不同频率的声音谐波响应不均匀，听起来感到失真；其次是这种回声太单调，每秒中的回声数目太少会导致声音的颤动，如果读者确实检验了 y1 的声音效果，就可以体验到这个问题。根据经验，回声的密度应该达到每秒 1000 个以上，听众才能有混响的感受。

为了解决第一个问题，人们就采用幅特性为常数的全通滤波器，比如传递函数为(9.6.5)式的全通滤波器，效果就比较好。

$$H(z)=\frac{\alpha+z^{-R}}{1+\alpha z^{-R}} \qquad |\alpha|<1 \tag{9.6.5}$$

为了对付第二个问题，需要采用多个迟延量 R 不同的 IIR 滤波器，模仿不同距离的反射物体，并将它们以并联或串联结构的方式组合起来等措施，现代的数字式家用音响放大器中都广泛地使用这些技术。

9.6.2 频域处理方法

把分别录制的各种乐器或歌手的声音进行混合时，通常要由音乐工程师修改它们的频率响应。方法是让信号通过一个均衡器（equalizer)，其目的是使得这些声音的中频区(1.5~3kHz)居于峰值，以便在混合信号中表现出来。另外也需要通过“扩大”或“削减”在此频率范围外的信号，以修正低频和高频的信号之间的关系。通常是用许多一阶和二阶的参数可调的滤波器级联起来实现这个功能的。在这里，滤波器的结构选择的主要要求之一是调整方便，最好是调一个参数只影响一个应用指标，而且可调参数要少。

一阶均衡器的可调滤波器结构图如图 9.6.2(a)，它的输入到两个输出端的传递函数分别为

$$H_{LP}(z)=0.5[1-A_1(z)],\ H_{HP}(z)=0.5[1+A_1(z)] \tag{9.6.6}$$

其中：

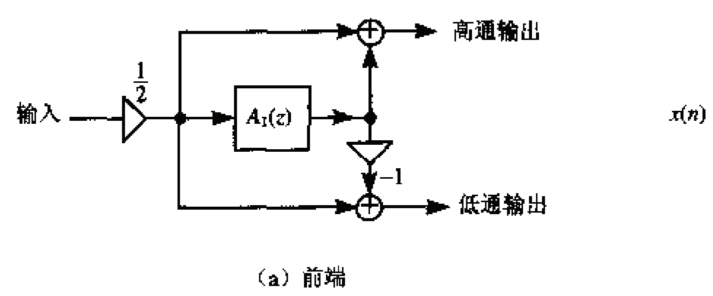
$$A_1(z)=\frac{\alpha-z^{-1}}{1-\alpha z^{-1}} \qquad |\alpha|<1 \tag{9.6.7}$$

再把两个输出端信号加权求和，就可以灵活地构成均衡滤波器。如把低频输出乘以 K 与高频输出相加，如图 9.6.2(b)，就称为低频均衡滤波器；其传递函数为

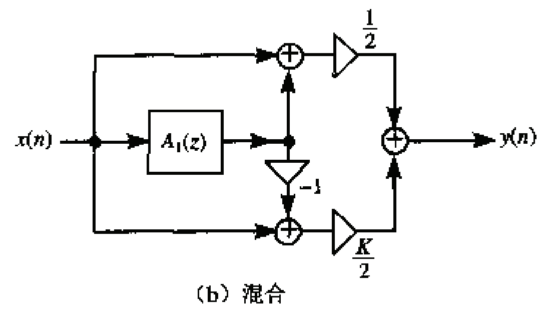
$$G_1(z)=0.5\cdot K\cdot[1-A_1(z)]+0.5\cdot[1+A_1(z)]=0.5(K+1)+0.5(1-K)A_1(z) \tag{9.6.8}$$

在 α 和 K 取不同数值时它们的频率响应可以用如下的 MATLAB 程序 hc962 计算。

```
K=2.5;alpha=0.7;
b1=[alpha,-1]; a1=[1,-alpha];
b = polyadd(0.5*(1+K)*a1,b1);
[H,w]=freqz(b,a1);
semilogx(w,abs(H))
```



(a) 前端



(b) 混合

图 9.6.2 频率均衡器的结构

得出的曲线如图 9.6.3 和图 9.6.4 所示。调节 K 用以改变低频端幅特性的提升程度，而调节 α 则用来控制低通滤波器的边界频率。

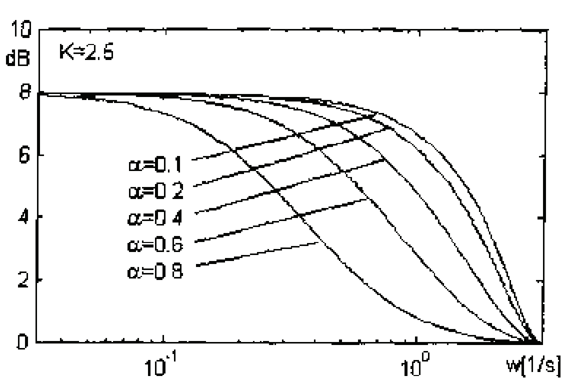


图 9.6.3　音乐信号的一阶低频均衡滤波器特性

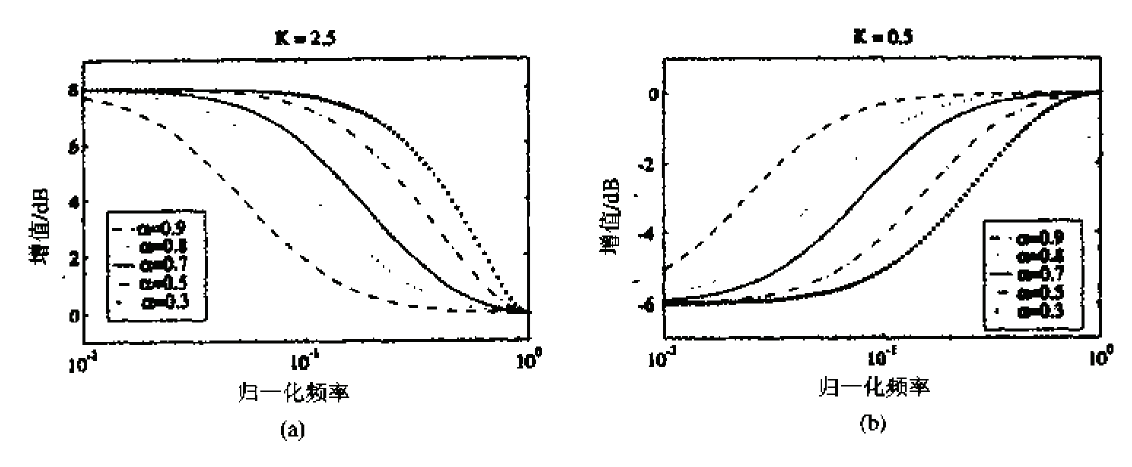


图 9.6.4　设定 K 改变 α 时一阶低频均衡器特性

如把高频输出乘以 K 与低频输出相加，就称为高频均衡滤波器，如公式(9.9.9)。请读者自行分析它的幅特性和参数调节的影响。

$$H_1(z)=0.5\bullet[1-A_1(z)]+0.5\bullet K\bullet[1+A_1(z)] \tag{9.6.9}$$

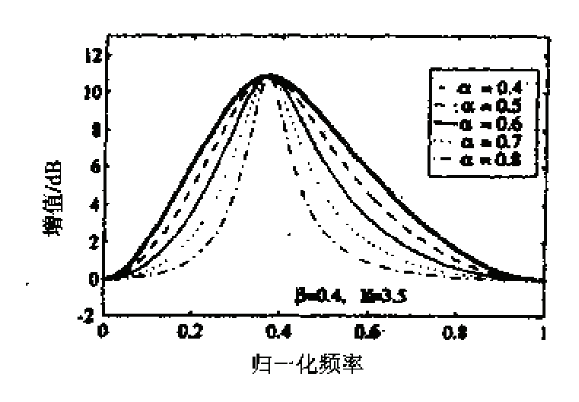
二阶均衡器也采用图 9.6.1 的结构图，不同点仅仅在于把其中的 $A_1(z)$换成 $A_2(z)$。

$$A_2(z)=\frac{\alpha-\beta(1+\alpha)\,z^{-1}+z^{-2}}{1-\beta(1+\alpha)\,z^{-1}+\alpha z^{-2}} \qquad |\alpha|<1 \tag{9.6.10}$$

因此把 $A_2(z)$ 替换了 $A_1(z)$后，公式(9.6.6)到(9.6.9)全部适用。因此可以写出

$$H_2(z)=0.5K[1-A_2(z)]+0.5[1+A_2(z)] \tag{9.6.11}$$

它的频率响应见图 9.6.5。与一阶均衡器的主要区别在于它有一个谐振峰，因此有三个可调参数。谐振峰的位置取决于 β，其幅度取决于 K，而它的宽度取决于 α。运用本书介绍的滤波器知识和 MATLAB 程序，读者不难自己计算出这些波形。实际的音乐播放器中的均衡器就采用多个调谐在不同频率上的二阶均衡器级联组成的。



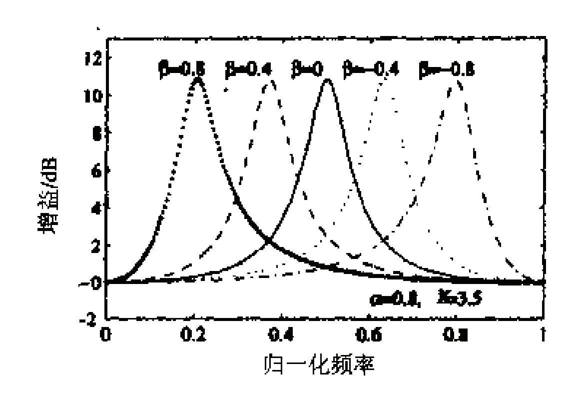


图 9.6.5 二阶均衡器的频率特性与参数的关系

9.7 变采样率数字滤波

前面所讨论的信号处理的各种方法都是把采样频率 F_s 视为固定值，即在一个数字系统中只有一个采样频率。但在实际系统中，经常会遇到采样率的转换问题，即要求一个数字系统能工作在“多采样率”状态。在数字系统越来越普及的情况下，各个数字系统都有自己不同的标准，它们之间的衔接也不可避免地要遇到这个问题。例如：

（1）在数字音响系统中，目前就使用了三种不同的采样频率：广播中采用的 32 kHz，声音光盘(CD)中采用的 44.1 kHz 和数字音频磁带(DAT)中采用的 48 kHz。

（2）在数字电视系统中，对 NTSC 制式的图像采集速率标准为 14.3181818MHz，而对 PAL 制为 17.734475MHz。对亮度信号采样率为 13.5MHz，对色度信号的采样率为 6.75MHz，两者又相差一倍。这些标准的差别就要求数字电视演播室系统工作在多采样率状态。

（3）在数字电话系统中，传输的信号既有语音信号，又有传真信号，甚至有视频信号，这些信号的频率成分相差甚远。所以，该系统应具有多种采样率，并根据所传输的信号自动完成采样率转换。

以上所列举的几个方面都是希望能对采样率进行转换，或要求数字系统工作在多采样率状态。近年来，建立在采样率转换基础上的“多采样率数字信号处理”已成为数字信号处理学科中的主要内容之一。

直观地想，变换采样率的最简单方法是：首先将以采样率 F_1 采集的数字信号进行 D / A 转换，变成模拟信号，再按采样率 F_2 进行 A / D 变换，这就实现了从 F_1 到 F_2 的采样率转换。但这样较麻烦，且易使信号受到损伤，所以在实用的变采样率系统中，改变采样率并不经过模拟信号，而完全是在数字域实现的。

采样率转换通常分为“抽取(decimation)”和“插值(interpolation)”。抽取是降低采样率(所以在英文中也称为 down-sampling)以去掉多余数据样本的过程，而插值则是提高采样率(up-sampling)以增加数据样本的过程。变采样率是一个非线性问题，它涉及相当复杂的数学推导，通常不可能在本科阶段掌握它的设计问题。本书主要介绍抽取和插值的一般概念，一方面可以加深对采样定理的理解，同时可以帮助读者了解更多的应用问题，开阔思路。

9.7.1 信号的整数倍内插

整数倍内插是在已知的相邻两个采样点之间插入 $L-1$ 个样本点。由于这些样本点并非已知的值，所以关键问题是如何求出这 $L-1$ 个样本值。前提是不经过模拟域，要直接在数字域

完成这个插值。

用数字方法实现整数内插的步骤如下：先在已知采样序列 $x(n_1T_1)$ 的相邻两个样点之间等间隔插入 $L-1$ 个零值样本点，假如其输出为 y，则输入输出关系为

$$y(n)=\begin{cases} x(n/L) & n=0,\pm L,\pm 2L,\cdots \\ 0 & n=\text{其他值} \end{cases}$$

式中 L 为大于 1 的整数，称为插入因子。然后经 $H(z)$ 进行低通数字滤波，把滤波得到的结果代入这些位置，得到内插的结果。这种内插方案的框图如图 9.7.1 所示。

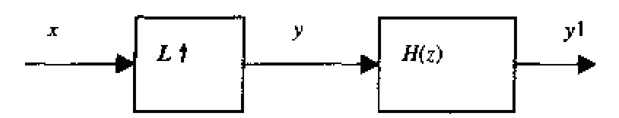


图 9.7.1　用数值方法提高采样率的框图

图 9.7.2 中左边的(a)、(b)、(c)、(d)、(e)子图是时域波形，(a)为原始输入序列 x，(b)是插入了 $L-1$ 个样本点的单位脉冲序列 p。(c)、(d)依次对应于图 9.7.1 中后两个信号：在相邻的 x 样本中插入 $L-1$ 个零值点后的序列 y 和经过数字低通滤波后的输出序列 y1。(e)的波形是把模拟量直接用高频率采样后的理想序列，用作参考比较。

右边的(f)、(g)、(h)、(i)、(j)是左边波形的频谱。频率轴是用模拟频率 Ω 单位为（1/ s）来标注的，以便找到采样率提高前后的对应关系。图 9.7.2 是用 MATLAB 程序 hc971 画成的，其中设输入为 $x=e^{-0.5t}$，原始采样频率为 Fsx=2 Hz，插入倍数为 L=3，x 序列长度为 Nx=10。低频滤波器任意地选择了例 7.3.3 中用布莱克曼窗函数设计的长度为 11 的 FIR 滤波器，其系数 b 可以运行 hc733 后查得，该程序的核心语句如下；

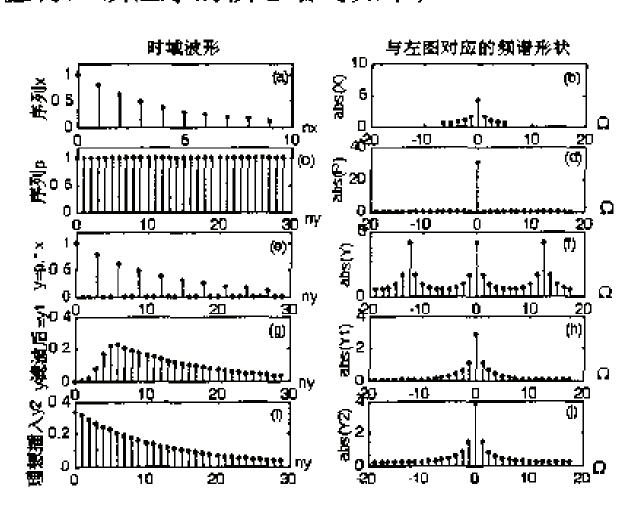


图 9.7.2　整数倍内插处理过程中的序列时域波形和对应的频谱特性

```
nx=0:Nx-1;                              % 设定x的自变量向量
x=exp(-alpha*nx*T);                     % 给出x序列的值
Ny=Nx*L;ny=0:Ny-1;                      %  设定y的自变量向量
p=ones(1,L*Nx);                         % 生成频率为L*Nx的单位脉冲序列p
y=zeros(1,L*Nx);                        % 先y序列初始化（全置零）
y(1:L:L*Nx)=x;                          % 实行内插，给出y序列在x对应点处的值
b=[0,0.0019,0.0203,0.0772,0.1589,0.2,0.1589,0.0772,0.0203,0.0019,0]; %滤波器系数
y1=filter(b,1,y);                       % 用例7.3.3的滤波器进行滤波
y2=exp(-alpha*ny*T/L)/L;                % 原始函数用高采样率的序列
X=fftshift(fft(x));                     % 求x频谱，并移到对称位置
```

```
Y=fftshift(fft(y));              % 求y 频谱，并移到对称位置
P=fftshift(fft(p));              % 求p 频谱，并移到对称位置
Y1=fftshift(fft(y1));            % 求y1 频谱，并移到对称位置
Y2=fftshift(ftt(y2));            % 求y2 频谱，并移到对称位置
nxm=floor(nx-Nx/2+0.5);          % 生成对零频率对称的 x 频率位置序列
nym=floor(ny-Ny/2+0.5);          % 生成对零频率对称的 Y 频率位置序列
绘图语句略去。
```

我们来分析整个整数倍插入过程中的信号变换关系。在由 x 变为 y 时，波形没有变，但采样率提高了，这相当于把序列 x 与一个单位样本序列 p 相乘，该 p 的采样频率 Fsy 为 x 的采样频率 Fsx 的 L 倍，因此，相乘生成的 y 序列采样频率与 p 相同，因为大部分点上 x=0，故 y 也为零。只是隔 L-1 个点有一个非零脉冲，其值等于该时刻的 $x(n)$。时域的相乘对应于频域的卷积，因此 y 的频谱 Y 相当于把原来 x 的频谱 X 与频谱 p 卷积，p 的采样频率比 x 高 L 倍，所以其频谱 p 是间隔宽了 L 倍的脉冲串。与 X 卷积的结果是把 X 平移复制 L-1 次。由于采样频率提高了 L 倍，所以其奈奎斯特频率也提高了 L 倍（注意在图上频谱曲线在横坐标方向所占的区域就是奈奎斯特频率区域），实际上 x 和 y 波形相同，频谱 Y 和 X 形状也是相同的，只是 Y 的奈奎斯特频段比 X 扩展了 L 倍。从频率域来看，要从 Y 取出原始频谱的问题变得容易了。因为现在的基本频谱在奈奎斯特频段中占的宽度只是原来的 $1/L$，所以很容易用低通滤波器滤去所有的镜像频率而保持原有的频谱。特别是因为这个滤波器的输入输出都是数字序列，可以方便地用 FIR 数字滤波器来实现过渡带很窄的低通滤波，远远优于模拟滤波器。这里我们用的是很初级的滤波器，经过数字滤波后的序列就是 y1。它的频谱就是列在图 9.7.2 的第四行子图中的 abs(Y1)。虽然用的滤波器性能很差，但显然它的频谱就是从 abs(Y) 图中基本滤去了镜像频谱的结果。

9.7.2 信号的整数倍抽取

整数倍抽取是在已知序列 x 中每隔 M-1 个采样点取出一个样本点，组成新的序列 y。因此 y 序列中的所有值都是已知的，可以直接写出下列表达式。

$$y(n_y)=x(n_yM)$$

也就是说，输入序列中下标为 M 的整倍数的点上的 x 都被取入 y 序列，而其他的 x 序列中的样本都被舍去不用了。用 MATLAB 语言来表达即为：

```
Ny=ceil(Nx/M);nx=0:Nx-1; ny=0:Ny-1;
y=x(0:M:Nx)          % 等距离地从 M 个 x 中抽取一个样本给 y 赋值
```

表面上看起来，抽取的数学关系较插入简单。但由于采样率的降低，造成的频率泄漏可能变得相当严重，以致抽取出的序列中引入了混叠或其他干扰信息。要解决好这个问题，在进行抽取之前，必须对原来的序列进行预先滤波，或称为“抗泄漏滤波”。它的框图如图 9.7.3。注意本节中信号命名的规则，取原来采样频率的信号称为 x, x1…改变了采样频率的信号命名为 y, y1…

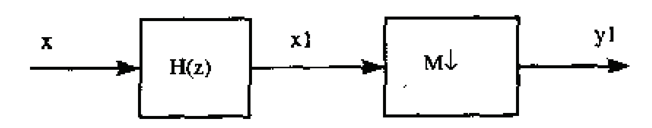


图 9.7.3　用数值方法降低采样率的框图

应该按照框图的次序和变量命名规则来编写它的 MATLAB 程序 hc972，先用上面同样的滤波器滤波得到 $x1$，再抽取得到 $y1$。所以其核心语句为：

```
nx=0:Nx-1; x=exp(-alpha*nx*T);           % 给出x序列的值
b=[0,0.0019,0.0203,0.0772,0.1589,0.2,0.1589,0.0772,0.0203,0.0019,0];  %
x1=filter(b,1,x);                        %用例733的滤波器系数进行滤波，得出x1
Ny=ceil(Nx/M);ny=0:Ny-1;
y1=x1(1:M:Nx);                           % 实行抽取，给出y1序列在x1对应点处的值
```

所有其他程序语句与 hc971 基本相同。在假定输入为 $x = e^{-0.5t}$，原始采样频率为 Fsx=5Hz，插入倍数为 M=5，x 序列长度为 Nx=100 的条件下，程序运行得到的波形见图 9.7.4。

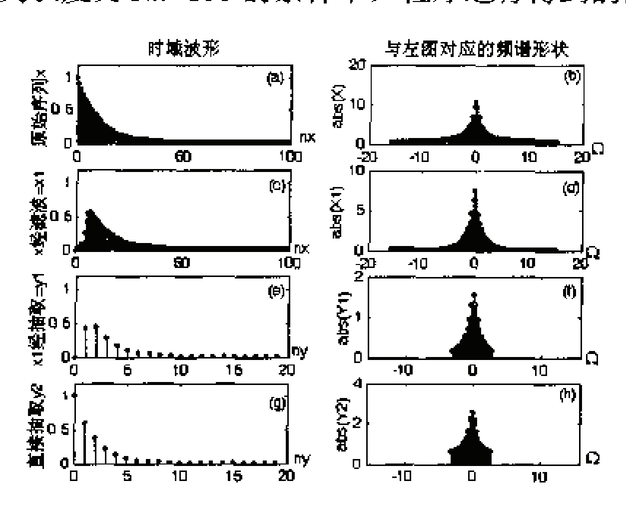


图 9.7.4　整数倍抽取处理过程中的序列时域波形和对应的频谱特性

从图中左边一列看出抽取过程的时域波形，指数下降的输入信号 x 经过低通滤波成为波形 $x1$，然后经过抽取而降低了采样率成为 $y1$，最下面的图表示不经过滤波直接抽取的结果。再从右边的频谱图上可以看出滤波的必要性。直接抽取波形的频谱 abs($Y2$)在奈奎斯特频率处（也就是频谱图的左右边界 $\pm\pi$ 处）有很高的幅度，这意味着存在着很大的频率泄漏。而经过滤波再抽取的信号频谱 abs($Y1$)在该处的幅度就小得多，如果用一个好一点的滤波器，结果会更加理想。所谓理想，就是它能够更好地复现 abs(X)中对应的低频分量。有兴趣的读者可以从本程序运行的数值结果中去检验。

9.7.3　分数倍变采样率和相应的 MATLAB 函数

如果采样频率的变换不是简单的整数倍抽取或整数倍内插，但变换前后的采样频率的比率满足两个整数之比 L/M，则不难想像，可以通过先作 L 倍内插后作 M 倍抽取方法来实现，也可以把次序倒过来，先抽取后内插。其滤波器设计的当然有不同的要求和方法。

MATLAB 信号处理工具箱提供了一个变采样率函数 resample，用它可以把变采样率的全过程在一步中完成。这个变采样率函数的典型调用格式为 y=resample(x,L,M)。其中 L 为采样率提高的倍数，M 为采样率降低的倍数，L 和 M 必须取整数。自然也可以想像，如果输入两个互为质数的 L 和 M 值，则就可以实现分数倍(L/M)的采样率变换了。

由前面的分析知道，在变采样率过程中，必须要一个适当的滤波器。这个函数中有默认的滤波器设计，它是按等波动最优化原理设计的，因此可以为用户省去设计滤波器的工作。当然，也可以由用户规定数字滤波器作为输入的选项，这时调用格式中要加第四个输入变元，

那就是 FIR 滤波器的系数向量。resample 函数的其他调用方法，请用 help 命令查阅。

9.7.4 整数倍抽取和内插的应用举例

在数字电话系统中，采样频率取为 F=8 kHz，希望传输尽量接近 4 kHz 的音频带宽。但送话器发出的信号 $x(t)$ 的带宽比 4 kHz 大很多。因此，在 A/D 变换之前要对它进行模拟预滤波，以防止采样后发生频率混叠失真。这就要求该滤波器在 4 kHz 处幅频特性很低，而为了使能传送的信号频带尽量宽，又要求该滤波器的通带接近于 4 kHz，给过渡带宽留的裕度只有几百 Hz，这样的要求对用模拟方法实现的低通滤波器是很难做到的。

为了降低对模拟预滤波器的技术要求，新方案先用较高的采样率进行采样，比如取采样率 F_{sx}=16 kHz，经过 A/D 后，再经 D=2 倍抽取，把采样率降低至 8 kHz。这时，模拟预滤波器的过渡带可以从 4 kHz 到 12 kHz，这就容易设计了。此时问题难点变成了抽取时的滤波器设计技术。因为这是在数字域的滤波，用 FIR 结构的数字滤波器不难设计成线形相位和过渡带很窄的特性，这种方案最终可有效地利用通信带宽，并增加信号数据传送量。

9.8 稀疏天线阵列设计

在雷达、声呐、超声成像和地震信号处理中广泛使用线性相位天线阵列。在实际中往往需要去除掉其中某些阵元，这就称为稀疏阵列。可以发现，等间隔分布的线性相位天线阵元所产生的远场方向图和 FIR 滤波器的频率响应在数学模型上非常相似。这就使我们可以根据方向图的要求来设计稀疏阵列。本节将简单地探讨这个问题，其目的在于扩大读者的思路。因为直到目前为止，本书讨论的信号处理都是以时间作为自变量的。而在本例中自变量却是一维的空间位置。从本例可以看到，利用数学模型的相似性，时域的信号处理理论和方法可以推广到空域的信号处理中。

图 9.8.1 所示为 N+1 个等间隔配置的方向性相同的阵元。阵元间距为 d，因此它们在基线上排列的坐标为 $x(n)=n\times d$，$0\leqslant n\leqslant N$。则在与基线垂直方向的远场辐射方向图可表示为

$$P(u)=\sum_{n=0}^{N}h(n)\mathrm{e}^{\mathrm{j}[2\pi(u/\lambda)d]n} \tag{9.8.1}$$

其中 $h(n)$第 n 个阵元的复数激励源，也称为复数权函数。而 $u=\sin\theta$，θ 为与基线的法线方向的夹角。如图 9.8.1 所示。

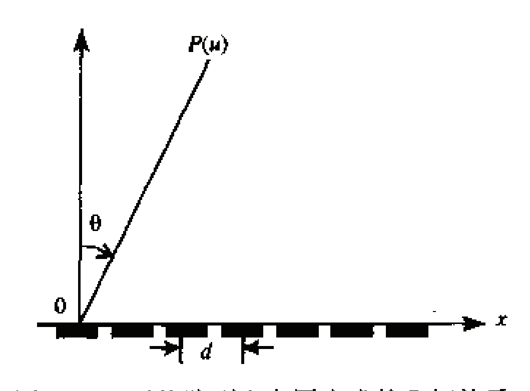


图 9.8.1　天线阵列方向图生成的几何关系

由此我们可以把 $P(u)$ 看做 $h(n)$ 与频率变量 $w=2\pi(u/\lambda)d$ 构成的离散傅里叶变换。对于均匀激励的阵列，$h(n)$=常数 a，如果 $d\leqslant\lambda/2$,那就可以避免栅格波瓣的出现。通常选择 $d=\lambda/2$,这时 w 的取值范围为 $-\pi\sim\pi$。从(9.8.1)式看出，$P(u)$ 的表示式与长度为 N+1 的 FIR 滤波器的频率响应的表示式相同。因此 FIR 滤波器的设计方法可以用于在给定方向图指标来设计阵列天线阵。因为最常见的情况是 $h(n)=1$，它就和一个矩形窗的频率响应相仿。

如果从此阵列中去掉某些阵元就形成了稀疏阵列。这时某些阵元间的间距将大于 $\lambda/2$,从而导致旁瓣幅度的增加和栅格波瓣的出现。为了减小这些不需要的副瓣，可以适当地调整收发阵元的位置。在超声扫描器中，通常用一组发射器和一组接收器组成一个双向辐射的有效方向图。它是由发射方向图与接收方向图的卷积形成的，设计这样的方向图就比较简单些。

如果天线阵是由一个发射阵元和 16 个非稀疏的接收阵元组成,那实际上仍然等价于非稀疏阵列，这时共需要 17 个阵元。如果用稀疏接收阵列（或稀疏发射阵列），就可以节省阵元的数目。比如，令发射阵列有两个阵元，而接收阵列有八个阵元，排列如下：

$$h_T(n)=[1\ \ 1],\ h_R(n)=[1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]$$

其中，0 表示去除的阵元。则有效方向图函数将由两者的卷积构成，即

$$h_{eff}(n)=h_T(n)\otimes h_R(n)=[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1] \tag{9.8.2}$$

因此只用了十个阵元就达到了前面 17 个阵元的效果。

可以用 MATLAB 计算 FIR 滤波器频率响应的函数来计算这个收发阵列的有效方向图。采用的程序 hc981 中的核心语句如下：

```
ht1=[1,1];hr1=[1,0,1,0,1,0,1,0,1,0,1,0,1,0,1]; % 给定稀疏收发阵元
he1=conv(ht1,hr1);                               %有效阵列等于收发阵列的卷积
[Hr1,w]=freqz(hr1,1,'whole');                    % 求稀疏接收阵列的频率特性（方向图）
[He1,w]=freqz(he1,1,'whole');                    % 求有效收发阵列的频率特性（方向图）
plot(w-pi,fftshift(abs(Hr1))/16,'-.',w-pi,fftshift(abs(He1))/16)
% 把频率轴的中心移到原点画图，除以 16，是为把零频处的幅特性数字为 1
legend('稀疏接收阵列的方向图','收发双向有效方向图')
```

程序运行的结果如图 9.8.2 所示。

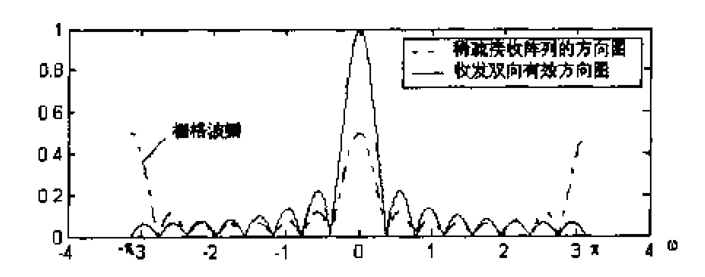


图 9.8.2　稀疏天线收发阵列卷积形成的有效方向图（用 FIR 滤波器频谱算法）

按照这个思路继续分析，可以找到更经济的方法，如果用 8 个阵元稀疏排列如下两式，也能够达到同样的效果。

$h_T(n)=[1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1],\ h_R(n)=[1\ 0\ 1\ 0\ 1\ 0\ 1]$　或

$h_T(n)=[1\ 1\ 1\ 1],\ h_R(n)=[1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1]$

以上方法的本质是用一个稀疏阵列来“填补”另一个稀疏阵列的空位。其结果是产生一个等效幅度全为 1 的激励源，相当于一根矩形窗。矩形窗的旁瓣是比较大的，会造成较大的栅格波瓣。可以用类似的概念进行插补，使得收发天线的等效激励源得到更平滑的分布，以

减小栅格波瓣。例如，给出如下的收发阵列：

$$h_T(n)=[1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1],\ h_R(n)=[1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]$$

它们卷积的结果是一个三角形的有效激励函数。

$$h_{eff}(n)=[1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 3\ 3\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1] \tag{9.8.3}$$

这就像三角窗那样，可以大大减小旁瓣。如果在阵列最边缘的阵元可以采用小于 1 的加权函数，则有效激励函数可以进一步平滑。例如，令

$$h_T(n)=[1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1],\ h_R(n)=[0.5\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0.5]$$

则得到的有效激励函数为

$$h_{eff}(n)=[0.5\ 0.5\ 1\ 1\ 1.5\ 1.5\ 2\ 2\ 2\ 2\ 2\ 2\ 1.5\ 1.5\ 1\ 1\ 0.5\ 0.5] \tag{9.8.4}$$

读者可自行分析画出它们的辐射方向图。

9.9　结束语

数字信号处理技术正以前所未有的速度在发展，它的发展是多方面的，在这里做一个扼要的介绍：

（1）首先是应用领域的拓宽。许多过去用模拟信号处理的设备，现在都力图用数字信号处理来代替。包括通信、语音处理、雷达和声呐信号处理、广播和电视技术、生物和医学信号处理、地震和地理信息、交通和工业控制等等。出现了数万种用于信号处理的专用集成电路芯片(ASIC for DSP)，使得数字信号处理的速度飞速提高，而其成本却不断降低。表 9.9.1 给出了 DSP 芯片应用的一些主要领域，这当然还是不完全的，只是提供一个概念。

（2）数字信号处理应用的发展提出了多种多样的复杂功能以及越来越高技术指标，推动了新的理论不断出现。本书讨论的只限于一维信号，要处理图形和图像就涉及二维信号处理，虽然它的基础是一维信号处理，但究竟更加复杂，而且带来许多新的更难的问题；在三维空间成像、虚拟现实、数字视频、数字电影和动画等又使信号处理进入了三维。在理论方面，它早就涉及了统计理论、非线性理论等，现在又出现了模糊集合、小波变换、神经元网络、还有各种谱估计理论等。所以十多年前，大学本科的信号处理课还是选修课，后来成了必修课。但新内容太多，又必须加 DSP 芯片软硬件设计课程，在研究生阶段还要设“高等信号处理”课；即使如此，也还很难跟上形势的发展。

（3）信号处理应用的发展推动了它的研制手段和工具的发展。除了 DSP 芯片的性能日新月异之外，作为一种科学计算软件，MATLAB 为这些数字信号处理系统提供了快捷的计算和仿真工具，主要为非实时的顶层设计使用。一般是用来研究总体方案和算法的可行性。现在除了信号处理工具箱(Signal)之外，它还有许多工具箱是与信号处理密切相关的。例如符号运算工具箱(Symbolic)、系统仿真工具箱(Simulink)、用于仿真的数字信号处理模块库(DSP Blockset)、定点运算模块库(Fix-Point Blockset)、统计处理工具箱(Statistcs)、图像处理工具箱(Image Processing)，模糊集合工具箱（Fuzzy Logic）、神经元网络工具箱(Neural Network) 、小波变换工具箱(Wavelet)等等。其他还有通信系统工具箱（Communication）、通信模块库（Communication Blocksets）等等，也都是分析信号处理问题时可能用到的工具。

在顶层设计得到验证后，再与实际的 DSP 芯片相结合，考虑它的速度和实时运行的可能性。在本课程中，我们只着重于前面部分，即原理和算法的部分，所以用 MATLAB 就足以解决问题了。从原理和算法向 DSP 芯片过渡也是一个相当麻烦的过程。除了要了解许多硬件

的知识外，在软件上从 MATLAB 语言转到芯片的汇编语言也要费很大的功夫。为了解决这个问题，Mathworks 公司提供了与 TI 公司和 Motorola 公司的 DSP 芯片的接口软件。从原理上说，用 MATLAB 和 Simulink 验证过的系统，就可以通过这个软件编译成为特定规范的 C 语言，这种 C 语言程序能够再编译为特定的 DSP 芯片的汇编语言。这两个工具箱分别称为 Developer's Kits for TI DSPs 和 Motorola DSP Developer's Kits。

因此，在本课程中我们把信号处理的基本原理和解决它的工具 MATLAB 结合起来，希望能够为读者今后的发展提供一个广阔的前景。

表 9.9.1　常见数字信号处理(DSP)芯片应用领域一览

通　用　DSP	图 形 图 像	仪 器 仪 表
数字滤波 卷积运算 相关运算 希尔波特变换 FFT 自适应滤波 加窗 波形生成	三维旋转 机器人视觉 图像传送和压缩 模式识别 图像增强 全息处理 工作站 动画/数字地图	频谱分析 函数发生 模式匹配 地震处理 暂态分析 数字滤波 锁相环
声学和语音	控　　制	军　　用
声音处理 语言编码 语声识别 说话人识别 语言增强 语声合成 文字朗读	磁盘光盘控制 伺服控制 机器人控制 激光打印机控制 发动机控制 电机控制	保密通信 雷达信号处理 声呐信号处理 图像处理 导航 导弹制导 无线电频率调制
通　　信		汽　　车
回声对消 ADPCM 编码传送 数字程控交换机 线路复用 Repeater 信道多路 调制解调器（MODEM） 自适应均衡 DTMF 编码 / 解码 数据加密	蜂窝通信 可视电话 数字语言 插值 X-25 信包转发器 视频会议 扩频通信	发动机控制 振动分析 防滑刹车 自适应驾驶控制 全球定位 汽车导航 语声命令 数字无线电 移动通信
消费电子	工业控制	医学应用
雷达测量 电力工具 数字音响和电视 音乐合成 教育和玩具	工业机器人 数控 安全出入 电力线路监测	助听器 病房监护 超声仪器 诊断和手术工具 假肢开发 生命监护

复习思考题

9.1 根据第 1 章中提出的信号处理的基本任务，联系学习本书中做过的应用题型，看是否都能用 sptool 工具来解决，如何解决？

9.2 试证明信号量化后产生的误差（看做噪声）的均方根为$\sigma=\frac{\Delta}{\sqrt{12}}=\frac{\Delta}{2\sqrt{3}}$。

9.3 有没有办法计算经过压扩器按图 9.2.1 处理后的信号量化误差噪声？应该怎么考虑？从图 9.2.3 能否得出噪声减小的定量评价？

9.4 为什么系数量化可以用相对量化的方法？而信号量化要用绝对量化的方法？

9.5 对相对量化的子程序 brqtize 逐条作出解释。用一个包含三个不同数量级元素的数组代入，分析它运算中出现的中间结果，说明它的原理。

9.6 为什么人们特别关心系数量化对滤波器极点位置的影响？

9.7 双音拨号系统比老式电话的脉冲拨号系统有什么优越性？尽量列举。如果不用数字信号处理而用模拟信号处理，设想它的组成框图和主要元件。

9.8 从双音拨号系统中分析为什么 DFT 可以作为一种滤波器来应用。为什么它采用了 Goertzel 算法求 DFT 而没有用 FFT?能否比较两者的运算复杂性？

9.9 用 FFT 实现一个双频拨号系统。

9.10 为什么对周期信号进行频谱分析时，截断的长度应该尽量是周期的整倍数？如果输入信号没有明显的周期性，截断的长度应该长还是短？各有什么优缺点？

9.11 怎样提高频率分辨率最有效？增加 FFT 点数还是增加信号长度？怎样减少截断所造成的高频频谱。

9.12 在通信线路上消除回波是一个重要技术命题。如果带有回声的信号模型如(9.6.1)式。要消除其中的回声部分，也就是要恢复 $x(n)$的原来信号形式，是否可以做到？应该怎样做？

9.13 分别设想用模拟方法和数字方法实现均衡器的两个方案。数字均衡器比模拟均衡器有哪些优点？

9.14 变采样率的运算也是离散序列的一种基本运算，但它属于非线性运算。请说明为什么它是非线性运算。

9.15 序列信号经过 L=2 的插入后，它的频谱与原信号相比，发生了什么变化？请列举出变化的几项内容。先插入后滤波是为了减小哪一项变化？

9.16 序列信号经过 M=2 的抽取后，它的频谱与原信号相比，发生了什么变化？请列举出变化的几项内容。先滤波后抽取是为了减小哪一项变化？

9.17 试比较空域信号处理与时域信号处理的相同点和不同点。

习题

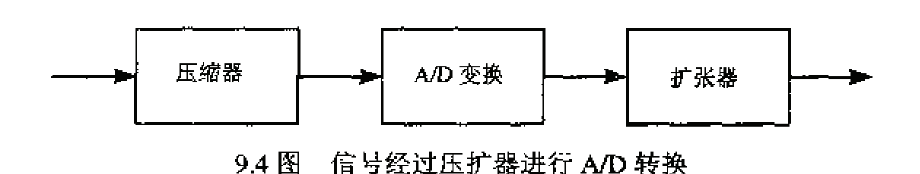
9.1 将文件 dajiahao.dat 中“大家好”的波形 x 放入 sptool 中，测量其波形的长度和最大最小幅度，与 MATLAB 工作空间的数组进行比较检验。测量其有效发声区的长度，把此有效发声区的数组设置为另一变量 $x1$，计算出它的长度，并把它也同时存入文件 dajiahao.dat 中。

9.2 运行程序 hc841，取第 8 章例 8.4.1 中的低通数字滤波器（分子系数向量 bd1，分母系数向量 ad1），导入 sptool 中。将上述 x 信号加入此滤波器，观测其输出信号的幅度变化，并听它的声音。若将此滤波器进行修改（用 sptool 界面中的【Edit】按钮）把其通带从约 2000 Hz

降到 500 Hz，再让 x 信号通过它，结果有何变化，试讨论之。

9.3　把例 8.4.1 中的其他数字滤波器(分子系数向量 bd2,bd3, bd4, 分母系数向量 ad2, ad3, ad4）导入 sptool 中。将上述 x 信号加入这些滤波器，听其输出信号的声音。各有何特征？如果在导入滤波器时，取的采样频率为 10000 Hz，做这个实验时有没有问题？能否不重新导入而解决问题？为什么？

9.4　设输入信号范围为 0～5 V，量化为四位二进制。采用压扩器连接，如下图所示。



9.4 图　信号经过压扩器进行 A/D 转换

编写程序求出它的输入输出特性、绝对量化误差和绝对量化误差，并画出图 9.2.2 右子图。用计算数据说明加了压扩器后，A/D 变换器对于大信号和小信号有基本相同的相对误差，并说明它的最小量化步长比不用压扩器减小了多少倍。

9.5 令幅度为 4V 的正弦信号通过幅度范围为±5V，精度为三位二进制（不算符号位）的 A/D 变换器，画出它的输出波形。若经过题图 9.4 所示的压扩器进行 A/D 变换，画出其波形。如果其他参数都不变，输入信号幅度减小为 1V，则经过和不经过压扩器有何区别？

9.6 将"大家好"的信号送入幅度范围为±1V，精度为三位二进制的的 A/D 变换器，听它的输出声音；若把信号减小 5 倍，即乘以 0.2，听输出声音的变化。然后将 A/D 变换器与压扩器连接如题 9.4 图，做同样的实验，从中可以得出什么结论？

9.7 比较绝对量化和相对量化子程序，为什么前者比后者要多一个输入变元？

9.8 将例 8.4.1 中带通数字滤波器的系数 bd3 和 ad3 量化为八位二进制（不算符号位），在同一张图上画出量化前后的频率响应，进行比较。问将量化位数增加到几位？才可以忽略量化的影响。

9.9 题 9-8 中，如果把直接结构的系数 bd3 和 ad3 转换为二阶级联形式的参数，问将量化位数取到几位？就可以忽略量化的影响。

9.10 在检测 DTMF 信号程序中，规定所取序列的长度为 N=205，在 F_s=8 kHz 时，这相当于 25ms。如果想减小这个长度，使 N=100，问程序应如何修改，才能正常工作。

9.11 修改程序 hc941，使它能适用于 N=50~400 的广阔范围，每次运行时 N 按提示由键盘输入。问 N 减低到多少时就会出错，为什么？讨论 N 的现有裕度及设置这样大裕度的原因。

9.12 考虑抗语音干扰时，即要同时考虑二次谐波检测时，程序应如何编写？

9.13 一个语言信号的采样率为 12 kHz，取其中长度为 256 的一段进行频谱分析。问

(a) 如果进行 256 点的 DFT，问 DFT 样本之间的分辨率为多少？

(b) 如果要使 DFT 分辨率达到 16 Hz，应该怎样做？

(c) 用什么方法能得到 128 Hz 的分辨率，且使 DFT 的长度为最小。

9.14 一个有限带宽 F_m 的连续实信号 $g_a(t)$用频率 F_s 进行采样，$F_s \geqslant 2F_m$。然后对采样后的序列进行 R 点 DFT。问

(a) 设 Fm=6kHz，问 R 和 F_s 该取多少，以保证其 DFT 具有 2.6Hz 的分辨率而且没有频率泄漏。

(b) 设 Fm=3kHz，如果 R 必须取为 2 的幂，以便用 FFT 进行计算。则 R 该取多少，以

保证具有 3 Hz 的分辨率而且所取的采样频率不会造成频率泄漏。

9.15 考虑长度为 64 的序列为 $x(n)=A\cos(2\pi f_1 n/64)+B\cos(2\pi f_2 n/64)$, $0\leqslant n\leqslant 63$。已知其 64 点 DFT 中，除了 k=15,27,37 和 49 外，其他 k 值处的频谱样本均等于零。若$|X(15)|=32$, $|X(27)|=16$，求出 $x(n)$的准确表达式。

9.16 一个信号的频谱包括两个数字频率为$f1$ 和$f2$ 的正弦分量，$f1<f2$。现在要用 DFT 对它进行分析。首先让此信号经过一个长度为 N 的窗函数 wd(n)，设 N=60，且正弦分量的第一个频率为$f1$=0.25，用一个 $R(R>N)$点的 DFT，试确定它所能分辨出的第二个频率$f2$ 的最小值。对以下的窗函数，分别给出答案。

(a) 矩形窗　(b) 哈明窗　(c) 汉宁窗　(d) 布莱克曼窗

9.17 设输入信号中有 50Hz 市电干扰，要用带阻滤波器将它滤除 95%。而对（50±5）Hz 以外的频率影响不大于 1%，试确定输入序列的最小长度。

9.18 设输入信号中有 50 Hz 和 52 Hz 两个分量，要用频谱分析的方法加以区分，相互泄漏不超过 10%，问输入信号的采样频率 F_s 如何选择，序列的长度 N 应如何选择？是否需要加窗、加什么窗好？

9.19 将例 9.6.1 中生成回波信号的 FIR 滤波器导入 sptool 中进行观察。说明它的幅特性和相特性各有什么特点。对生成多次回波信号的 IIR 滤波器也做同样的分析。

9.20 将例 9.6.1 中的原信号和生成的回波信号导入 sptool 中进行观察。说明它的特点。

9.21 编写程序，画出一阶可调均衡器的调谐特性图 9.6.3。

9.22 编写程序，画出二阶可调均衡器的调谐特性图 9.6.5。

9.23 用 resample 函数，按同样的信号和内插参数，生成例 9.7.1 中的内插序列，画出它的时域和频域波形，并与书上的结果进行比较，说明其差别的原因。

9.24 用 resample 函数，按同样的信号和内插参数，生成例 9.7.2 中的抽取序列，画出它的时域和频域波形，并与书上的结果进行比较，说明其差别的原因。

9.25 求出加权激励函数为公式(9.8.3)和(9.8.4)时的波瓣特性，确定其最大旁瓣的值。

附录 A 信号处理工具箱函数表

版本 5.1 (R12.1) 06-Apr-2001

分　　类	函　数　名	功 能 说 明	首次在本书出现的章节号
滤波器分析	abs	幅值	3.2
	angle	相角	3.2
	freqs	模拟系统频率响应	3.3
	freqspace	为频率响应设定频率间隔	—
	freqz	数字滤波器频率响应	3.3
	freqzplot	频率响应绘制	—
	fvtool	滤波器可视化工具	—
	grpdelay	群迟延	7.3
	impz	脉冲响应(离散的)	2.5
	unwrap	修正相位,使其范围不限于主角±π	7.6
	zplane	画出离散的零极增益点	5.3
滤波器实现	conv	卷积	2.3
	conv2.	二维卷积	—
	deconv	解卷积	5.2
	fftfilt	重叠相加滤波器实现	3.7
	filter	滤波器实现	2.4
	filter2	二维滤波器实现	—
	filtfilt	零相位滤波	—
	filtic	确定滤波器原始条件	5.3
	latcfilt	格形滤波器实现	6.5
	medfilt1	一维的中值滤波	—
	sgolayfilt	Savitzky Golay 滤波器实现	—
	sosfilt	二阶环节(biquad)滤波实现	6.4
	upfirdn,	先高采样，后 FIR 滤波，再低采样	—
FIR 滤波器设计	convmtx	卷积矩阵	—
	cremez	复非线性相位等波动 FIR 滤波器设计	—
	fir1	基于窗函数 FIR 滤波器设计	7.3
	fir2	基于窗函数的任意响应 FIR 滤波器设计	7.4
	fircls	约束的最小二乘法任意响应滤波器设计	—
	fircls1	约束最小二乘法低通和高通滤波器设计	—
	firls	最小二乘法 FIR 滤波器设计	7.5
	firrcos	上升余弦 FIR 滤波器设计	—
	intfilt	插值 FIR 滤波器设计	—
	kaiserord	基于窗函数的 Kaiser 滤波器阶数选择	—
	remez	Parks McClellan 最适的 FIR 滤波器设计	7.5
	remezord	Parks McClellan 滤波器阶数估计	7.5
	sgolay	Savitzky Golay FIR 平滑滤波器设计	—

（续表）

分　类	函　数　名	功　能　说　明	首次在本书出现的章节号
IIR 数字滤波器设计	butter	巴特沃斯滤波器设计	8.6
	cheby1	切比雪夫-I 型滤波器设计	8.6
	cheby2	切比雪夫-II 型滤波器设计	8.6
	ellip	椭圆型滤波器设计	8.6
	maxflat	归一化的巴特沃斯低通滤波器设计	—
	yulewalk	耶鲁-沃克滤波器设计	—
IIR 滤波器阶数估算	buttord	巴特沃斯滤波器阶数选择	8.2
	cheb1ord	切比雪夫-I 型滤波阶数选择	8.2
	cheb2ord	切比雪夫-II 型滤波器阶数选择	8.2
	ellipord	椭圆型滤波器阶数选择	8.2
模拟低通滤波器原型	besselap	贝塞尔滤波器原型	—
	buttap	巴特沃斯滤波器原型	8.2
	cheb1ap	切比雪夫类型 I 滤波原型(带通波动)	8.2
	cheb2ap	切比雪夫类型 II 滤波器原型(带阻波动)	8.2
	ellipap	椭圆型滤波器原型	8.2
模拟低通滤波器设计	besself	贝塞尔模拟滤波器设计	—
	butter	巴特沃斯滤波器设计	8.6
	cheby1	切比雪夫类型 I 滤波设计	8.6
	cheby2	切比雪夫类型 II 滤波器设计	8.6
	ellip	椭圆的滤波器设计	8.6
模拟滤波器频带变换	lp2bp	低通向带通模拟滤波器变换	8.4
	lp2bs	低通向带阻模拟滤波器变换	8.4
	lp2hp	低通向高通模拟滤波器变换	8.4
	lp2lp	低通向低通模拟滤波器变换	8.4
滤波器离散化	bilinear	有预先修正选项的双线性变换	8.3
	impinvar	脉冲响应不变法模拟向数字的转换	8.3
线性系统变换	latc2tf	格形或者格形梯形向传递函数转换	6.5
	polystab	使多项式稳定	—
	polyscale	多项式根乘以倍率	—
	residuez	z 变换部分分式展开	5.2
	sos2ss	级联二阶环节向状态空间转换	6.6
	sos2tf	级联二阶环节向传递函数转换	6.4
	sos2zp	级联二阶环节向零极增益转换	6.6
	ss2sos	状态空间转换为二阶环节级联	6.6
	ss2tf	状态空间向传递函数转换	6.6
	ss2zp	状态空间向零极增益转换	6.6
	tf2latc	传递函数向格形或者格形梯形转换	6.5
	tf2sos	传递函数向级联二阶环节转换	6.4
	tf2ss	状态空间向传递函数转换	6.6
	tf2zp	传递函数向零极增益转换	5.3

（续表）

分　类	函　数　名	功 能 说 明	首次在本书内出现的章节
线性系统变换	zp2sos	零极增益向级联二阶环节转换	6.6
	zp2ss	零极增益向状态空间转换	6.6
	zp2tf	零极增益向传递函数转换	5.3
窗函数	bartlett	Bartlett 窗函数	7.3
	barthannwin	修正巴特利特-汉宁窗	—
	blackman	布莱克曼窗函数	7.3
	blackmanharris	最小四项 Blackman-Harris 窗函数	—
	bohmanwin	Bohman 窗函数	—
	chebwin	切比雪夫窗函数	—
	gausswin	高斯窗函数	—
	hamming	哈明窗函数	7.3
	hann	汉宁窗函数	7.3
	kaiser	凯泽窗函数	7.3
	nutallwin	Nuttall 最小四项 Blackman-Harris 窗函数	7.3
	rectwin	矩形窗函数	7.3
	triang	三角窗函数	7.3
	tukeywin	Tukey 窗函数	—
	window	窗函数引入	—
变换	bitrevorder	将输入交换成倒序排列	4.2
	czt	线性调频 z 变换	5.4
	dct	离散的余弦变换	—
	dftmtx	离散傅里叶变换矩阵	3.5
	fft	快速傅里叶变换	3.5
	fft2	二维快速傅里叶变换	—
	fftshift	交换矢量的一半	3.5
	goertzel	计算 DFT 的 goertzel 算法	9.4
	hilbert	Hilbert 变换	7.6
	idct	离散的逆余弦变换	—
	ifft	快速傅里叶逆变换	3.5
	ifft2	二维快速傅里叶逆变换	—
倒谱分析	cceps	复倒谱	—
	Icceps	逆复倒谱	—
	rceps	实倒谱和最小相位重建	—
统计信号处理和谱分析	cohere	相干函数	—
	corrcoef	相关系数	—
	corrmtx	自相关矩阵	—
	cov	协方差矩阵	—
	csd	互相关谱密度	—
	pburg	用 Burg 方法功率谱估计	—
	pcov	用协方差方法功率谱估计	—

（续表）

分　类	函　数　名	功 能 说 明	首次在本书出现的章节号
统计信号处理和谱分析	peig	用特征向量方法功率谱估计	—
	periodogram	周期谱图方法功率谱估计	—
	pmcov	用修改协方差方法的功率谱估计	—
	pmtm	用 Thomson 多带方法功率谱估计	—
	pmusic	用 MUSIC 方法的功率谱估计	—
	psdplot	绘制功率谱密度数据	—
	pwelch	用 Welch 的方法功率谱估计	—
	pyulear	用耶鲁-沃克 AR 方法的功率谱估计	—
	rooteig	用特征向量法作正弦频率功率谱估计	—
	rootmusic	用 MUSIC 法作正弦频率功率谱估计	—
	tfe	传递函数估计	—
	xcorr	互相关函数	—
	xcorr2	二维互相关	—
	xcov	协方差函数	—
参数建模	arburg	用 Burg 的方法 AR 参数的建模	—
	arcov	用协方差方法 AR 参数的建模	—
	armcov	用修改协方差方法 AR 参数的建模用	—
	aryule	用耶鲁-沃克方法 AR 参数的建模	—
	ident	参看系统辨识工具箱	—
	invfreqs	模拟滤波器向频率响应拟合	—
	invfreqz	离散的滤波器向频率响应拟合	—
	prony	Prony 离散滤波器拟合时间响应	—
	stmcb	Steiglitz McBride 迭代的 ARMA 建模	—
线性预测	ac2rc	自相关序列向反射系数转换	—
	ac2poly	自相关序列向预测多项式安排转换	—
	is2rc	反正弦参数向反射系数转换	—
	lar2rc	对数区域的比向反射系数转换	—
	levinson	Levinson Durbin 递归	—
	lpc	使用自相关方法的线性预测系数	—
	lsf2poly	线谱频率向预测多项式的转换	—
	poly2ac	预测多项式向自相关序列转换	—
	poly2lsf	预测多项式向线谱频率转换	—
	poly2rc	预测多项式向反射系数转换	—
	rc2ac	反射系数向自相关序列转换	—
	rc2is	反射系数向反正弦参数转换	—
	rc2lar	反射系数向对数区域比转换	—
	rc2poly	反射系数向预测多项式转换	—
	rlevinson	逆 Levinson Durbin 递归	—
	schurrc	Schur 算法	—

（续表）

分　类	函　数　名	功 能 说 明	首次在本书出现的章节号
多采样率信号处理	decimate	用低采样速度再抽样	9.7
	downsample	抽取输入信号	—
	interp	用更高采样速度再抽样	9.7
	interp1	通用一维插值透入(MATLAB 实现箱)	—
	resample	用新取样速度再抽样	9.7
	spline	三次样条内插	—
	upfirdn	先内插，后 FIR 滤波，再抽取	—
	upsample	对输入信号内插	—
波形产生	chirp	扫频的频率余弦发生器	2.6
	diric	Dirichlet(周期性 sinc)函数	2.6
	gauspuls	高斯高频脉冲发生器	—
	gmonopuls	高斯单脉冲发生器	—
	pulstran	脉冲序列发生器	2.6
	rectpuls	采样的非周期性的方波发生器	2.6
	sawtooth	锯齿函数	2.6
	sinc	Sinc 或者 sin(pi*x)/pi*x 函数	2.6
	square	方波函数	2.6
	tripuls	采样的非周期性的三角形发生器	2.6
	vco	压控振荡器	—
专门的运算	buffer	把一个矢量缓冲到数据桢的矩阵中去	—
	cell2sos	将单元阵列转换为二阶级联矩阵	—
	cplxpair	把矢量变为复共轭对	6.4
	demod	解调	—
	dpss	离散的扩展球形序列(Slepian 序列)	—
	dpssclear	从数据库去除离散的扩展球形序列	—
	dpssdir	离散的扩展球形序列数据库子目录	—
	dpssload	从数据库下载离散的扩展球形序列	—
	dpsssave	向数据库写入离散的扩展球形序列	—
	eqtflength	使离散时间传递函数分子分母系数等长	—
	modulate	为通信仿真调制	—
	seqperiod	在一个矢量中找到最小长度重复序列	—
	sos2cell	将二阶级联矩阵转换为单元阵列	—
	specgram	语言信号谱图	—
	stem	画出离散数据序列	1.1
	strips	画出条形图	—
	udecode	使输入二进制码均匀解码	—
	uencode	使输入信号均匀编码为 N 位二进制码	—
图形用户界面工具	fdatool	滤波器分析设计工具	8.6
	sptool	信号处理工具	9.2

（续表）

分　类	函　数　名	功 能 说 明	首次在本书出现的章节号
音频支持（这些函数不是工具箱函数，而是属于 MATLAB 基本部分）	sound	重放矢量成为声音	—
	soundsc	声音自动定标和重放矢量	—
	wavplay	使用视窗音频的输出装置重放声音	—
	wavread	读微软.wav 声音文件	1.3
	wavrecord	使用视窗音频的输入装置记录声音	—
	wavwrite	写微软 .wav 声音文件	—

附录 B　本书自编子程序索引表

序号	章节号	名　称	功　能	典型调用格式
1	7.2	amplres	由 FIR 滤波器系数求符幅特性	[A,w,type,L0]=amplres(h)
2	9.2	alawcom	A-律压缩器	y=alawcom(x, A, V)
3	8.3	bilinear0	模拟系数到数字系数双线性变换	[bd,ad] = bilinear0(ba, aa, Fs)
4	8.3	bilnrmap	双线性变换把 s 多项式变为 z 多项式	[bd,ad] = bilnrmap(ba, aa, Fs)
5	9.2	bqtize	二进制绝对量化程序	y=bqtize(x,N,V)
6	9.4	brqtize	二进制相对量化程序	y=brqtize(x,N)
7	3.6	circonv	循环卷积计算（不需调用自编函数）	y = circonv(x1,x2,N)
8	3.6	cirshift	循环移位计算	y=cirshift(x,m,N)
9	2.3	convwthn	带下标向量的卷积	[y,ny] = convwthn (x, nx, h, nh)
10	6.4	cplxcomp	复数极点 p1 变为 p2 后留数的序号	I=cplxcomp(p1,p2)
11	2.3	evenodd	序列分解为奇对称与偶对称两部分	[xe,xo,m]=evenodd(x,nx)
12	9.4	gfft	Goertzel 算法计算 DFT	X=gfft(x,N,k)
13	8.3	Impinvar0	脉冲响应不变法模拟到数字变换	[bd,ad]=impinvar0(ba,aa,Fs)
14	7.3	ideallp	理想低通滤波器脉冲响应函数	hd = ideallp(wc,N)
15	2.2.2	impseq	脉冲序列	[x,n] = impseq(n0,n1,n2)
16	9.2	mulawcom	μ-律压缩器函数	y= mulawcom(x,mu,V)
17	9.2	mulawexp	μ-律扩张器函数	y= mulawexp(x,mu,V)
18	4.2	mydft	用矩阵乘法算 DFT	y= mydft(x)
19	7.3	myfreqz	z 域频率响应计算的改进版本-	[db,mag,pha,grd,w] = myfreqz(b,a)
20	4.2	myditfft	MATLAB 语言编写的基 2 FFT 程序	y=myditfft(x)
21	6.4	par2tf	并联结构变为直接结构	[b,a] = par2tf(C,B,A)
22	6.4	parfilt	并联结构滤波器进行滤波	y = parfilt(C,B,A,x)
23	2.2.3	seqadd	序列相加函数	[y,n] = seqadd(x1,n1,x2,n2)
24	2.2.3	seqfold	序列折叠函数	[y,n] = seqfold(x,n)
25	2.2.3	seqmult	序列相乘函数	[y,n] = seqmult (x1,n1,x2,n2)
26	2.2.3	seqshift	序列平移函数	[y,ny] = seqshift(x,nx,n0)
27	2.2.2	stepseq	阶跃序列	[x,n] = stepseq(n0,n1,n2)
28	6.3	tf2fs	直接结构转换为频率样本结构	[C,B,A] = tf2fs(h)
29	6.4	tf2par	直接结构转换为并联结构	[C,B,A] = tf2par(b,a)
30	8.5	zmapping	z 域内数字滤波器频带转换函数	[bz,az] = zmapping(bZ,aZ,Nz,Dz)

附录C　例题程序索引表

序　号	名　称	功　　能	产生图号及调用的重要函数
1	hc111	绘制模拟信号，离散序列和数字序列	图 1.1.1
2	hc221	基本序列的生成	图 2.2.1
3	hc222	离散序列的合成	图 2.2.2
4	hc223	位置向量不同的序列的相加和相乘	图 2.2.3
5	hc231	序列的周期延拓	图 2.3.1
6	hc232	序列分解为奇偶序列	图 2.3.2
7	hc233	序列的相加和相乘及折叠	图 2.3.3
8	hc234	序列的折叠、移位和卷积	图 2.3.4
9	hc235	convwthn 序列的相关计算	—
10	hc236	序列的相关计算	图 2.3.5
11	hc251	卷积波形的图示	图 2.5.1
12	hc252	用向量-toeplitz 矩阵相乘求卷积	—
13	hc323	矩形序列的 DTFT 解析计算	图 3.2.1
14	hc324	四点序列的 DTFT 数值计算	图 3.2.2
15	hc325	对称序列及其移位后的 DTFT 比较	图 3.2.3
16	hc326	DTFT 线性性质检验	—
17	hc327	实序列及复序列的 DTFT 比较	图 3.2.4
18	hc328	DTFT 共扼性质检验	—
19	hc329	由理想低通滤波器频谱反求序列(IDTFT)	图 3.2.5
20	hc331	一阶离散系统的频率响应和输入输出	图 3.3.2，图 3.3.3
21	hc341	周期序列的离散傅里叶级数(DFS)	图 3.4.2
22	hc351	DTFT 和 DFT 及 FFT 的关系	图 3.5.1
23	hc352	补零增加序列长度提高视在分辨率	图 3.5.2
24	hc361	循环移位特性的图示	图 3.6.1
25	hc362	循环折叠特性的图示	图 3.6.2
26	hc363	实序列分解为循环奇序列和循环偶序列	图 3.6.3
27	hc364	复序列分解为循环奇序列和循环偶序列	图 3.6.4
28	hc371	循环卷积和线性卷积的关系	—
29	hc431	五点序列的 DFT 和补零频谱计算	图 4.3.1，图 4.3.2
30	hc433	矩形窗补零后的频谱计算	图 4.3.3
31	hc434	按给定误差自动选取序列长度的程序	—
32	hc435	序列截取长度对频谱的影响	图 4.3.4
33	hc451	采样周期 T 和数据长度 L 对频谱计算的影响	图 4.5.1
34	hc451a	数据截断对频谱泄漏的影响	图 4.5.2
35	hc452	两个很靠近的频率分量的分辨率问题	图 4.5.3
36	hc453	Sinc 连续信号求频谱	图 4.5.4
37	hc453a	Sinc 连续信号求频谱（比较补零的结果）	图 4.5.4
38	hc454	截断的正余弦信号的频谱及补零问题	图 4.5.5
39	hc455	含直流分量的正弦信号的频谱	图 4.5.6

（续表）

序 号	名 称	功 能	产生图号及调用的重要函数
40	hc456	*T,N,L* 变化时频点集的对应关系	图 4.5.7
41	hc461	归一化频率 IDTFT 点数对序列混叠的影响	图 4.6.1
42	hc462	非归一化频率 IDTFT 点数对序列混叠的影响	图 4.6.2
43	hc463	IDTFT 求连续时间信号	图 4.6.3
44	hc463a	IDTFT 点数对连续信号时间混叠的影响	图 4.6.3
45	hc471	用 FFT 计算序列能量	—
46	hc511	具有同样 z 变换的左序列和右序列	图 5.1.1
47	hc523	极点留数法求 z 反变换（实极点）	—
48	hc527	极点留数法求 z 反变换（复数极点）	—
49	hc528	用逆 z 变换求差分方程全解（有初始条件）	—
50	hc529	用逆 z 变换求差分方程全解（有输入）	—
51	hc531	用 z 变换求系统输出	图 5.3.2
52	hc532	z 平面零极点分布和三维幅特性图	图 5.3.3，图 5.3.4
53	hc542	小于单位圆上频率的频谱分析	图 5.4.3
54	hc543	非单位圆上螺旋线 Chirp-z 的频谱分析	图 5.4.6
55	hc551	理想低通滤波器的脉冲响应及因果性	图 5.5.3
56	hc552	截取长度及对称性对频率特性的影响	图 5.5.5
57	hc561	混合信号通过一阶数字滤波器的输出	图 5.6.3
58	hc621	用信号流图法计算滤波器的传递函数	—
59	hc642	直接结构转换为并联结构	—
60	hc643	直接、级联和并联组合结构的传递函数	—
61	hc651	全零格型滤波器的计算	—
62	hc654	用信号流图求格型梯形滤波器的传递函数	—
64	hc661	正余弦信号产生器的信号流图分析	—
65	hc662	正余弦信号产生器的状态空间分析	—
66	hc721	幅频特性与符幅特性的区别	图 7.2.1
67	hc722	线性相位 FIR 滤波器的类型及符幅特性	图 7.2.3
68	hc731	矩形窗的符幅特性和累积符幅特性	图 7.3.3
69	hc732	三角窗的符幅特性和累积符幅特性	图 7.3.5
70	hc733	用三种窗函数设计低通滤波器	图 7.3.7
71	hc734	窗函数设计 FIR 低通滤波器	图 7.3.8
72	hc735	用凯泽窗函数设计低通滤波器	图 7.3.9
73	hc736	用布莱克曼窗函数设计带通滤波器	图 7.3.10
74	hc741	用频率样本法设计 FIR 滤波器	图 7.4.3
75	hc741a	理想频率响应	图 7.4.2
76	hc741b	用频率样本法取不同样本数时的幅频特性	图 7.4.4
77	hc742	过渡带有不同样本数时的幅频特性比较	—
78	hc742a	过渡带设一个样本时的试凑程序	—
79	hc742b	过渡带设两个样本时的设计程序	图 7.4.5
80	Hc743	用 fir2 函数生成限带波形	图 7.4.6

（续表）

序　号	名　称	功　　能	产生图号及调用的重要函数
81	hc751	用离散最小二乘法设计 FIR 滤波器	图 7.5.1
82	hc752	（逐步逼近）等波动法设计低通滤波器 1	图 7.5.4
83	hc752a	（逐步逼近）等波动法设计低通滤波器 2	图 7.5.5
84	hc752b	（逐步逼近）等波动法设计低通滤波器 3	图 7.5.5
85	hc753	等波动法设计低通滤波器	图 7.5.6
86	hc753a	同 hc753,内置判断语句	图 7.5.6
87	hc754	等波动法设计带通滤波器	图 7.5.7
88	hc761	加窗后数字微分器的幅频特性	图 7.6.1
89	hc762	希尔波特变换器的设计	图 7.6.4
90	hc821	巴特沃斯模拟滤波器的系统函数	—
91	hc822	巴特沃斯模拟滤波器设计	图 8.2.3
92	hc823	切比雪夫 I 型模拟滤波器设计	图 8.2.6
93	hc824	切比雪夫 II 型模拟滤波器设计	图 8.2.8
94	hc825	椭圆型模拟滤波器设计	图 8.2.10
95	hc831	脉冲响应不变法模拟系统与数字系统的比较	图 8.3.3
96	hc832	脉冲响应不变法由巴特沃斯设计数字滤波器	图 8.3.4
97	hc834	双线性变换法模拟系统与数字系统的比较	图 8.3.6
98	hc835	双线性变换法由巴特沃斯变数字滤波器	图 8.3.7
99	hc836	双线性变换法由切比雪夫 I 变数字滤波器	图 8.3.8
100	hc837	双线性变换法由切比雪夫 II 变数字滤波器	图 8.3.9
101	hc838	双线性变换法由椭圆变数字滤波器	图 8.3.10
102	hc841	由低通模拟变换为其他类型模拟频带的变换	图 8.4.4
103	hc841a	例 8.4.1 采样周期 T 减小 10 倍	—
104	hc841b	例 8.4.1 采样周期 T 加大 10 倍	—
105	hc851	由低通数字变换为高通数字频带的变换示例	图 8.5.4
106	hc861	切比雪夫 I 型带阻滤波器直接设计	图 8.6.1
107	hc921	mu 和 A-律压缩器曲线绘制	图 9.2.1
108	hc922	不加压缩扩张器的量化误差	图 9.2.2
109	hc922a	不加和加压缩扩张器的量化误差对比	图 9.2.2
110	hc923a	不加和加压缩扩张器对双音信号的影响	比较声音
111	hc923b	不加和加压缩扩张器对语声信号的影响	比较声音
112	hc931	系数量化对 FIR 滤波器频率特性的影响	图 9.3.1
113	hc932	系数量化对 IIR 滤波器极点分布的影响	图 9.3.2
114	hc941	DTMF 生成和检测程序（用自编函数）	图 9.4.1，图 9.4.2 及声音
115	hc941a	DTMF 生成和检测程序（用工具箱函数）	图 9.4.1，图 9.4.2 及声音
116	hc951	正余弦信号的频谱分析	图 9.5.1
117	hc952	信号长度和 DFT 长度对单频频谱分析的影响	图 9.5.2
118	hc952a	信号长度和 DFT 长度对双频频谱分析的影响	图 9.5.2
119	hc953	窗函数对频谱分析的影响	图 9.5.3，图 9.5.4
120	hc954	短时间傅里叶分析及时频分析	图 9.5.5
121	hc961	回声信号的生成	图 9.6.2

（续表）

序　号	名　称	功　　能	产生图号及调用的重要函数
122	hc962	一阶可调缓冲滤波器的频率响应	图 9.6.3
123	hc971	提高采样率(内插)数字序列的生成	图 9.7.2
124	hc972	降低采样率(抽取)数字序列的生成	图 9.7.4
125	hc981	稀疏收发天线阵列的方向图生成	图 9.8.2